软件系统分析与设计

**PaperIsAllYouNeed学术成果分享平台**

**架构调研报告**

|  |  |  |  |
| --- | --- | --- | --- |
| **项目组成员信息** | | | |
| **小组编号** | **17** | | |
| **小组名称** | **请填写小组名称** | | |
| **小组联系人** | **张津赫** | | |
| **学号** | **姓名** | **联系方式** | **本次实践中主要承担的工作内容** |
| 19373802 | 刘志一 | 13986777478 | 撰写报告 |
| 19231179 | 申玉啸 | 18810955230 | 架构调研，资料收集 |
| 19231251 | 奚望 | 13817322661 | 架构调研，资料收集 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

2021

目录

[一、淘宝网简介 3](#_Toc87261290)

[二、应用场景及难题 3](#_Toc87261291)

[三、核心架构设计 4](#_Toc87261292)

[3.1 高性能 4](#_Toc87261293)

[3.1.1 Oracle及拓展 4](#_Toc87261294)

[3.1.2 引入 NoSQL 数据库和搜索引擎等技术 5](#_Toc87261295)

[3.1.3 TFS文件系统 5](#_Toc87261296)

[3.1.4服务化 6](#_Toc87261297)

[3.2高扩展性 7](#_Toc87261298)

[3.2.1应用无状态 7](#_Toc87261299)

[3.2.2云平台部署 8](#_Toc87261300)

[3.3高可靠性 8](#_Toc87261301)

[3.3.1 监控、预警系统 8](#_Toc87261302)

[3.3.2 配置统一管理 9](#_Toc87261303)

[四、总结 9](#_Toc87261304)

[五、参考文献 9](#_Toc87261305)

# 一、淘宝网简介

淘宝网是亚太地区较大的网络零售、商圈，由阿里巴巴集团在2003年5月创立，，是中国深受欢迎的网购零售平台，拥有近5亿的注册用户数，每天有超过6000万的固定访客，同时每天的在线商品数已经超过了8亿件，平均每分钟售出4.8万件商品。截止2011年年底，淘宝网单日交易额峰值达到43.8亿元，创造270.8万直接且充分就业机会。2019年12月12日，《汇桔网·2019胡润品牌榜》发布，淘宝以3000亿元品牌价值排名第四。随着淘宝网规模的扩大和用户数量的增加，淘宝也从单一的C2C网络集市变成了包括C2C、分销、拍卖、直供、众筹、定制等等多种电子商务模式在内的综合性零售商圈。

在软实力方面，淘宝网也做得很好。淘宝网建立了可靠的实名认证体系。一旦淘宝发现用户注册资料中主要内容是虚假的，淘宝可以随时终止与该用户的服务协议，利用网络信息共享优势，建立公开透明的信用评价系统。同时，淘宝拥有自己的支付平台，支付宝的推出，解决了买家对于先付钱而得不到所购买的产品或得到的是与卖家在网上的声明不一致的劣质产品的担忧；同时也解决了卖家对于先发货而得不到钱的担忧。

# 二、应用场景及难题

通过公开数据可以看到淘宝网每天有超过6000万的固定访客，同时每天的在线商品数已经超过了8亿件，平均每分钟售出4.8万件商品。遇到消费旺季，大促销活动如过年季，双11等，这项数据还会膨胀数倍。商品条目有的还包含了商品演示视频，商品详情页还有长长的介绍图片……先不论购买的行为，单单是存储这些商品的数据所消耗的资源都是难以想象的。维护海量数据、处理海量订单、客户客服聊天等等功能无不要求淘宝网具有高性能，高并发，巨大规模的数据存储。

除了整个系统需要满足高性能需求之外，还得同时具备高可扩展性(高可伸缩性)。“双十一”是淘宝的传统，每逢打折促销的购物节，活跃客户量和订单数将会剧增，服务器负载和压力更会变大。这就使得这个系统必须具备高可伸缩性，在并发高峰来临之前，能够通过简单的加机器或者使用云化技术来保证系统可以正常运转。

在具备上述两个特点的同时，还得具备高可靠性和安全性。首先这样数量级的资源和并发单靠一台机器是不可能实现，必须采用集群来分散压力。而在集群中，必须防备机器故障，单台机器故障之后需要有其他机器作为替补，并且还必须在短时间将故障修复，恢复或清理掉损失的数据。同时，淘宝与支付行为直接挂钩，需要保证用户的银行信息，密码信息等不被窃取和泄露，保证用户支付的安全性。

为了用户操作的舒适性和便捷性，在非功能系统需求的实现上，淘宝也应做到尽善尽美。例如设计美观和清晰的ui界面，快速而准确的商品搜索功能，完善和负责的客服团队……这些功能也许不是淘宝团队技术方面的重点，但却是实打实地提高了用户体验，是架构设计时不可忽视的点。

# 三、核心架构设计

## 3.1 高性能

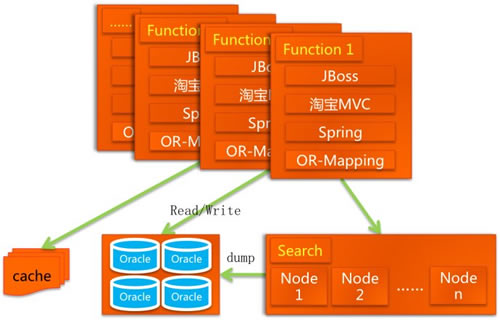
### 3.1.1 Oracle及拓展

淘宝海量的数据及高并发已经是当时的mySQL所解决不了的了，虽然换成了最先进的，处理并发最强的Oracle数据库。虽然暂时还能支持一亿PV的网站，但淘宝网的发展实在是太快了，到了十亿怎么办？到了百亿怎么办？淘宝不得不创造技术，解决这些只有世界顶尖的网站才会遇到的问题。

一台Oracle的处理能力是有上限的，它的连接池有数量限制，查询速度跟容量成反比。简单的说，在数据量上亿、查询量上亿的时候，就到它的极限了。要突破这种极限，最简单的方式就是多用几个Oracle数据库。但Oracle本身是一个封闭的系统，一个封闭的系统做扩展，不像分布式系统那样轻松。但淘宝的工程师凭借自己的研究实现了Oracle的拓展——分库分表，同时运用缓存和CDN技术。

分库分表从单一DB server,到Master/salve,再到垂直分区（分库），然后再到水平分区（分表，sharding），将烦人的商品详情属性存储到单独的一个表中。同时，访问这个页面的时候，数据全都是只读的（全部从数据库里面读出来，不写入数据库），如果把这些读操作从数据库里面移到内存（缓存）里，数据库将会多么的轻松。于是一位大神找到了一个基于 Berkeley DB 的开源的缓存系统，把很多不太变动的只读信息放了进去，形成了缓存系统。此外LVS 的创始人章文嵩博士带人搭建了淘宝自己的CDN网络。

这些杂七杂八的修改，我们对数据分库、加入缓存、加入 CDN、看起来没有章法可循，其实都是围绕着提高容量、提高性能、节约成本来做的，所形成的总体架构如下图。



**图1 Oracle拓展后的架构图**

### 3.1.2 引入 NoSQL 数据库和搜索引擎等技术

当数据库中的数据多到一定规模时，数据库就不适用于复杂的查询了，往往只能满足普通查询的场景。对于统计报表场景，在数据量大时不一定能跑出结果，而且在跑复杂查询时会导致其他查询变慢，对于全文检索、可变数据结构等场景，数据库天生不适用。因此需要针对特定的场景，引入合适的解决方案。如对于海量文件存储，可通过分布式文件系统 HDFS 解决。

对于 Key Value 类型的数据，可通过 HBase 和 Redis 等方案解决，对于全文检索场景，可通过搜索引擎如 ElasticSearch 解决，对于多维分析场景，可通过 Kylin 或 Druid 等方案解决。

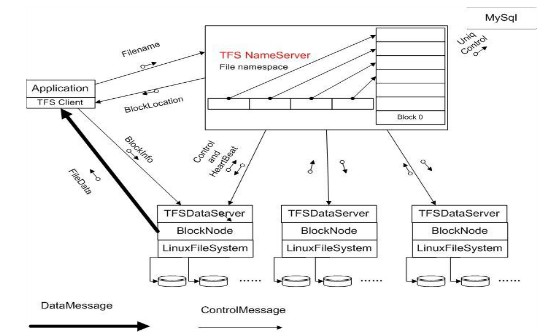
当然，引入更多组件同时会提高系统的复杂度，不同的组件保存的数据需要同步，需要考虑一致性的问题，需要有更多的运维手段来管理这些组件等。引入更多组件解决了丰富的需求，业务维度能够极大扩充，随之而来的是一个应用中包含了太多的业务代码，业务的升级迭代变得困难。

### 3.1.3 TFS文件系统

对于淘宝每个商品条目，其商品详情又长又多的图片无疑是存储中关键的一环。淘宝对图片存储的需求大概可以描述如下：

文件比较小；并发量高；读操作远大于写操作；访问随机；没有文件修改的操作；要求存储成本低；能容灾能备份。应对这种需求，显然要用分布式存储系统；由于文件大小比较统一，可以采用专有文件系统；并发量高，读写随机性强，需要更少的 IO 操作；考虑到成本和备份，需要用廉价的存储设备；考虑到容灾，需要能平滑扩容。

参照 GFS 并做了适度的优化之后，TFS 1.0 版的架构图如下：



**图2 TFS文件系统架构图**

从上面架构图上看：集群由一对 Name Server 和多台 Data Server构成，Name Server 的两台服务器互为双机，就是集群文件系统中管理节点的概念。

在这个架构中：

• 每个 Data Server 运行在一台普通的 Linux 主机上

• 以 block 文件的形式存放数据文件(一般64M一个block )

• block 存多份保证数据安全

• 利用 ext3 文件系统存放数据文件

• 磁盘 raid5 做数据冗余

• 文件名内置元数据信息，用户自己保存 TFS 文件名与实际文件的对照关系，使得元数据量特别小。

淘宝TFS文件系统在核心设计上最大的取巧的地方就在，传统的集群系统里面元数据只有 1 份，通常由管理节点来管理，因而很容易成为瓶颈。而对于淘宝网的用户来说，图片文件究竟用什么名字来保存实际上用户并不关心，因此TFS 在设计规划上考虑在图片的保存文件名上暗藏了一些元数据信息，例如图片的大小、时间、访问频次等等信息，包括所在的逻辑块号。而在元数据上，实际上保存的 信息很少，因此元数据结构非常简单。仅仅只需要一个 fileID，能够准确定位文件在什么地方。

由于大量的文件信息都隐藏在文件名中，整个系统完全抛弃了传统的目录树结构，因为目录树开销最大。拿掉后，整个集群的高可扩展性极大提高。实际上，这一 设计理念和目前业界的“对象存储”较为类似，淘宝网 TFS 文件系统已经更新到 1.3 版本，在生产系统的性能已经得到验证，且不断得到了完善和优化，淘宝网目前在对象存储领域的研究已经走在前列。

### 3.1.4服务化

越来越庞大与臃肿的淘宝将交易这个核心业务模块拆分出来了。原来的淘宝交易除了跟商品管理耦合在一起，也在支付宝和淘宝之间跳来跳去，跟支付宝耦合在一起，系统复杂，用户体验也很不好。淘宝把交易的底层业务拆出来叫交易中心TC（trade center），所谓底层业务是例如创建订单、减库存、修改订单状态等原子型的操作；交易的上层业务叫交易管理TM（trade manager），例如拍下一件普通商品要对订单、库存、物流进行操作，拍下虚拟商品不需要对物流进行操作，这些在TM里面完成。这个时候还有一个项目也在进行，就是淘宝商城，之前拆分出来的那些基础服务，给商城的快速构建，提供了良好的基础。

类目属性、用户中心、交易中心，随着这些模块逐步的拆分和服务化改造，在系统架构方面也积累了不少的经验。到 2008 年底淘宝做了一个更大的项目，把淘宝所有的业务都模块化，这是继 2004 年从 LAMP 架构到 Java 架构之后的第二次脱胎换骨，形成的体系架构如下。



**图3 服务化的架构图**

## 3.2高扩展性

### 3.2.1应用无状态

俗话说，一个系统的伸缩性的好坏取决于应用的状态如何管理。为什么这么说呢？试想一下，假如我们在session中保存了大量与客户端的状态信息的话，那么当保存状态信息的server宕机的时候，我们怎么办？通常来说，都是通过集群来解决这个问题，而通常所说的集群，不仅有负载均衡，更重要的是要有失效恢复failover,比如tomcat采用的集群节点广播复制，jboss采用的配对复制等session状态复制策略，但是集群中的状态恢复也有其缺点，那就是严重影响了系统的伸缩性，系统不能通过增加更多的机器来达到良好的水平伸缩，因为集群节点间session的 通信会随着节点的增多而开销增大，因此要想做到应用本身的伸缩性，我们需要保证应用的无状态性，这样集群中的各个节点来说都是相同的，从而是的系统更好的水平伸缩。

上面说了无状态的重要性，那么具体如何实现无状态呢？此时一个session框架就会发挥作用了。淘宝已经具有了此类框架。淘宝的session框架采用的是client cookie实现，主要将状态保存到了cookie里 面，这样就使得应用节点本身不需要保存任何状态信息，这样在系统用户变多的时候，就可以通过增加更多的应用节点来达到水平扩展的目的。但是采用客户端cookie的方式来保存状态也会遇到限制，比如每个cookie一般不能超过4K的大小，同时很多浏览器都限制一个站点最多保存20cookie.淘宝cookie框架采用的是“多值cookie”， 就是一个组合键对应多个cookie的值，这样不仅可以防止cookie数量超过20，同时还节省了cookie存储有效信息的空间，因为默认每个cookie都会有大约50个字节的元信息来描述cookie。

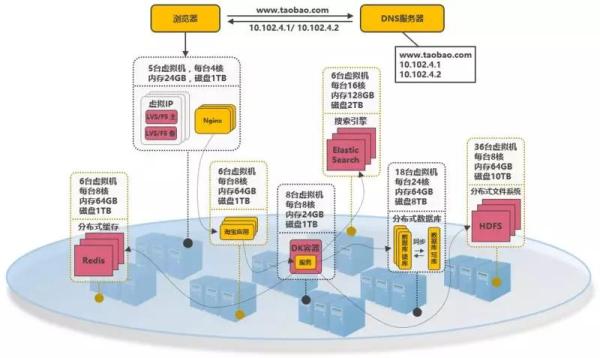
除了淘宝目前的session框架的实现方式以外，其实集中式session管理来完成，说具体点就是多个无状态的应用节点连接一个session 服务器，session服务器将session保存到缓存中，session服务器后端再配有底层持久性数据源，比如数据库，文件系统等等。

### 3.2.2云平台部署

系统可部署到公有云上，利用公有云的海量机器资源，解决动态硬件资源的问题。

在大促的时间段里，在云平台中临时申请更多的资源，结合 Docker 和 K8S 来快速部署服务，在大促结束后释放资源，真正做到按需付费，资源利用率大大提高，同时大大降低了运维成本。

所谓的云平台，就是把海量机器资源，通过统一的资源管理，抽象为一个资源整体，在之上可按需动态申请硬件资源(如 CPU、内存、网络等)，并且之上提供通用的操作系统。提供常用的技术组件(如 Hadoop 技术栈，MPP 数据库等)供用户使用，甚至提供开发好的应用，用户不需要关系应用内部使用了什么技术，就能够解决需求(如音视频转码服务、邮件服务、个人博客等)。



**图4 云平台部署架构图**

## 3.3高可靠性

### 3.3.1 监控、预警系统

对于大型的系统来说，唯一可靠的就是系统的各个部分是不可靠。因为一个大型的分布式系统中势必会涉及到各种各样的设备，比如网络交换机，普通PC机，各种型号的网卡，硬盘，内存等等，而这些东西都在数量非常多的时候，出现错误的概率也会变大，因此我们需要时时刻刻监控系统的状态，而监控也有粒度的粗细之分，粒度粗一点的话，我们需要对整个应用系统进行监控，比如目前的系统网络流量是多少，内存利用率是多少，IO，CPU的 负载是多少，服务的访问压力是多少，服务的响应时间是多少等这一系列的监控，而细粒度一点的话，我们就需对比如应用中的某个功能，某个URL的访问量是多少，每个页面的PV是多少，页面每天占用的带宽是多少，页面渲染时间是多少，静态资源比如图片每天占用的带宽是多少等等进行进一步细粒度的监控。因此一个监控系统就变得必不可少了。

有了监控系统以后，更重要的是要和预警系统结合起来，比如当某个页面访问量增多的时候，系统能自动预警，某台Server的CPU和 内存占用率突然变大的时候，系统也能自动预警，当并发请求丢失严重的时候，系统也能自动预警等等，这样以来通过监控系统和预警系统的结合可以使得我们能快速响应系统出现的问题，提高系统的稳定性和可用性。

### 3.3.2 配置统一管理

大型的分布式应用，一般都是有很多节点构成的，如果每次一个新的节点加入都要更改其它节点的配置，或者每次删除一个节点也要更改配置的话，这样不仅不利于系统的维护和管理，同时也更加容易引入错误。另外很多时候集群中的很多系统的配置都是一样的，如果不进行统一的配置管理，就需要再所有的系统上维护一份配置，这样会造成配置的管理维护很麻烦，而通过一个统一的配置管理可以使得这些问题得到很好的解决，当有新的节点加入或者删除的时候，配置管理系统可以通知各个节点更新配置，从而达到所有节点的配置一致性，这样既方便也不会出错。

# 四、总结

淘宝网从仅十人团队的创立开始，采用逐步优化，逐渐演变地形式，一步一个脚印地改善着它的架构。数据库从mySQL到Oracle，开发语言从PHP到Java，去IOE化，自研TFS文件系统和KV缓存系统，模块服务化，最后统一架构体系，底层的基础架构统一采用了阿里云计算平台，使用了阿里云计算服务，并通过阿里云服务提供的高可用特性，实现双机房容灾和异地机房单元化部署，为淘宝业务提供稳定、高效和易于维护的基础架构支撑。淘宝运用的技术功能不断升级，技术水平在不断进步，最后发展为了一家拥有高技术，有业界高端自研产品，自造轮子的优秀互联网公司。正因如此敢于变革，敢于创新，追求卓越的精神，淘宝才能成为中国网购第一大企。截至2014年底，淘宝网拥有注册会员近5亿，日活跃用户超1.2亿，在线商品数量达到10亿，在c2c市场，淘宝网占95.1%的市场份额，希望淘宝能在未来带来更先进的技术和更好的服务。

# 五、参考文献

[1] 风吹雨碎 《系统架构--淘宝网--淘宝网采用什么技术架构来实现网站高负载的》2012-3

<https://blog.csdn.net/gaolinwu/article/details/7338330?spm=1001.2101.3001.6650.3&utm_medium=distribute.pc_relevant.none-task-blog-2~default~BlogCommendFromBaidu~default-3.highlightwordscore&depth_1-utm_source=distribute.pc_relevant.none-task-blog-2~default~BlogCommendFromBaidu~default-3.highlightwordscore>

[2] Hank\_黄 《淘宝架构框架》2015-5 https://blog.csdn.net/yellowd1/article/details/45980269/

[3] 我思知我在 《淘宝技术架构演进之路--精华版》2021-5 <https://blog.csdn.net/qq_32828253/article/details/116763631?spm=1001.2101.3001.6661.1&utm_medium=distribute.pc_relevant_t0.none-task-blog-2%7Edefault%7ECTRLIST%7Edefault-1.highlightwordscore&depth_1-utm_source=distribute.pc_relevant_t0.none-task-blog-2%7Edefault%7ECTRLIST%7Edefault-1.highlightwordscore>

[4] 欧仕华 《千万级并发下，淘宝服务端架构如何演进？》2019-6 https://developer.51cto.com/art/201906/597895.htm