本文侧重介绍淘宝网后台的图片存储系统架构、包括 TFS 集群文件系统，以及前端处理服务器架构。

解决海量并发小文件的系统噩梦

对于淘宝网这类型访问量极高的电子交易网站来说，对图片系统的要求和日常的照片分享完全不在一个级别。日常照片分享往往集中在几个有限的亲朋好友之间，访问量不会特别高，而淘宝网商铺中的商品照片，尤其是热门商品，图片的访问流量其实是非常大的。而且对于卖家来说，图片远胜于文字描述，因此卖家也格外看重图片的显示质量、上传时间、访问速度等等问题。根据淘宝网的流量分析，整个淘宝网流量中，图片的访问流量会占到 90% 以上，而主站的网页则占到不到 10%。

淘宝网电子商城首页截图，淘宝网的后端系统上保存着 286 亿多个图片文件，淘宝网整体流量中，图片的访问流量要占到 90%以上。且这些图片平均大小为 17.45KB，小于 8K 的图片占整体图片数量 61%，整体系统容量的 11%

与此同时，这些图片的存储与读取还有一些头疼的要求：例如，这些图片要求根据不同的应用位置，生成不同大小规格的缩略图。考虑到多种不同的应用场景以及改版的可能性，一张原图有可能需要生成 20 多个不同尺寸规格的缩略图。

淘宝整体图片存储系统容量 1800TB（1.8PB），已经占用空间 990TB（约 1PB）。保存的图片文件数量达到 286 亿多个，这些图片文件包括根据原图生成的缩略图。平均图片大小是 17.45K；8K 以下图片占图片数总量的 61%，占存储容量的 11%。

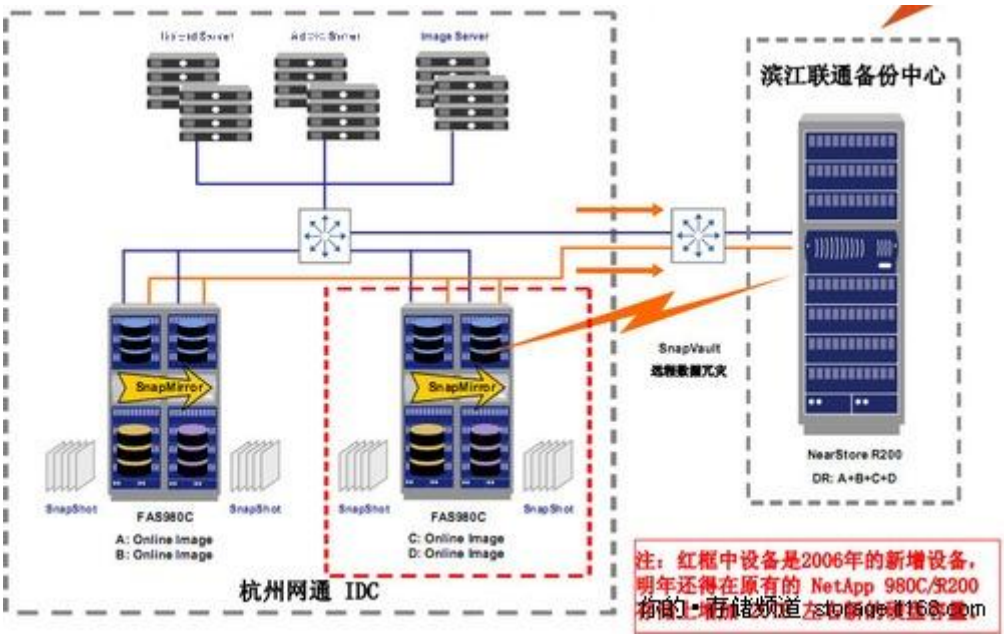
这就给淘宝网的系统带来了一个巨大的挑战，众所周知，对于大多数系统来说，最头疼的就是大规模的小文件存储与读取，因为磁头需要频繁的寻道和换道，因此在读取上容易带来较长的延时。在大量高并发访问量的情况下，简直就是系统的噩梦。

分析自主研发和商用系统的经济效益

淘宝网成立于 2003 年，在整个系统的构建和规划上

也做过相当多的尝试和探索。

下图是淘宝网 2007 年之前的图片存储系统。淘宝网之前一直采用的商用存储系统，应用 NetApp 公司的文件存储系统。随着淘宝网的图片文件数量以每年 2 倍(即原来 3 倍)的速度增长，淘宝网后端 NetApp 公司的存储系统也从低端到高端不断迁移，直至 2006 年，即时是 NetApp 公司最高端的产品也不能满足淘宝网存储的要求。



淘宝网 2007 年以前的图片存储系统架构图，由于淘宝网图片速度已每年 2 倍的速度增长，商用系统已经完全不能满足其存储需求，目前淘宝网采用自主研发的 TFS 集群文件系统来解决海量小图片的读取和访问问题。

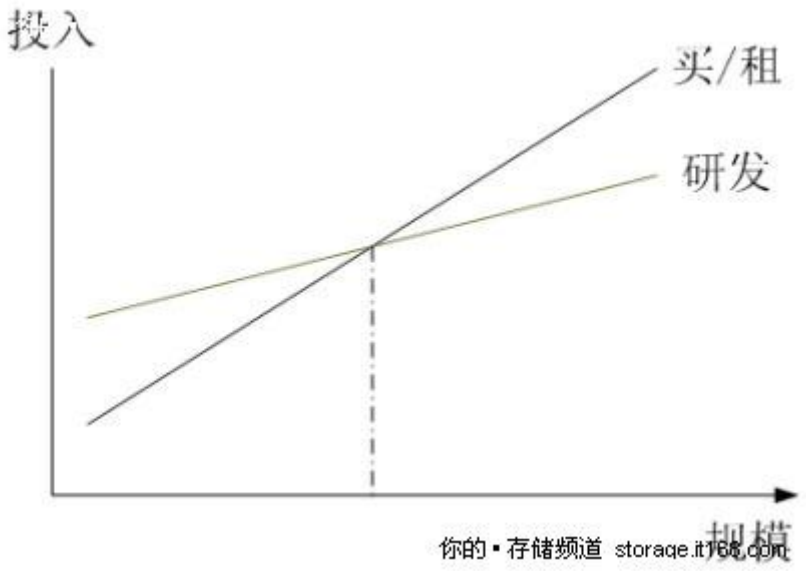
章文嵩博士在这里总结了几点商用存储系统的局限和不足：

首先是商用的存储系统没有对小文件存储和读取的环境进行有针对性的优化；其次，文件数量大，网络存储设备无法支撑；另外，整个系统所连接的服务器也越来越多，网络连接数已经到达了网络存储设备的极限。此外，商用存储系统扩容成本高，10T 的存储容量需要几百万￥，而且存在单点故障，容灾和安全性无法得到很好的保证。

谈到在商用系统和自主研发之间的经济效益对比，章文嵩博士列举了以下几点经验：

1. 商用软件很难满足大规模系统的应用需求，无论存储还是 CDN 还是负载均衡，因为在厂商实验室端，很难实现如此大的数据规模测试。

2. 研发过程中，将开源和自主开发相结合，会有更好的可控性，系统出问题了，完全可以从底层解决问题，系统扩展性也更高。



自主研发和采用商用系统的经济效益对比

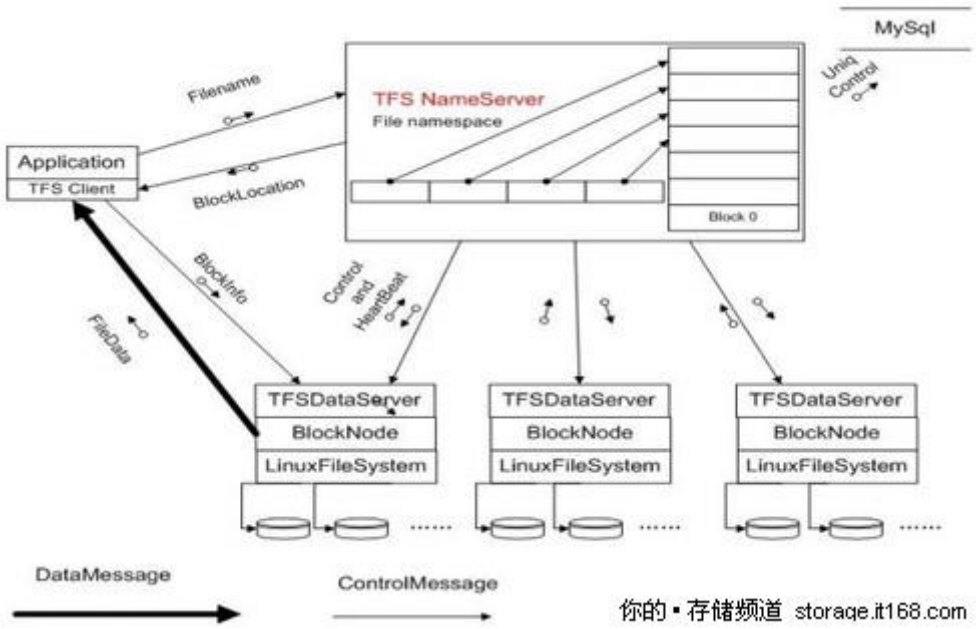
3. 在一定规模效应基础上，研发的投入都是值得的。上图是一个自主研发和购买商用系统的投入产出比对比，实际上，在上图的交叉点左边，购买商用系统都是更加实际和经济性更好的选择，只有在规模超过交叉点的情况下，自主研发才能收到较好的经济效果，实际上，规模化达到如此程度的公司其实并不多，不过淘宝网已经远远超过了交叉点。

4. 自主研发的系统可在软件和硬件多个层次不断的优化。

TFS 1.0 版本的集群文件系统

从 2006 年开始，淘宝网决定自己开发一套针对海量小文件存储难题的文件系统，用于解决自身图片存储的难题。到 2007 年 6 月，TFS(淘宝文件系统，Taobao File

System)正式上线运营。在生产环境中应用的集群规模达到了 200 台 PC Server(146G*6 SAS 15K Raid5)，文件数量达到上亿级别；系统部署存储容量: 140 TB；实际使用存储容量: 50 TB；单台支持随机 IOPS 200+，流量 3MBps。



淘宝集群文件系统 TFS 1.0 第一版的逻辑架构，TFS 最大的特点就是将一部分元数据隐藏到图片的保存文件名上，大大简化了元数据，消除了管理节点对整体系统性能的制约，这一理念和目前业界流行的“对象存储”较为类似。

图为淘宝集群文件系统 TFS 1.0 第一版的逻辑架构：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。

- 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。

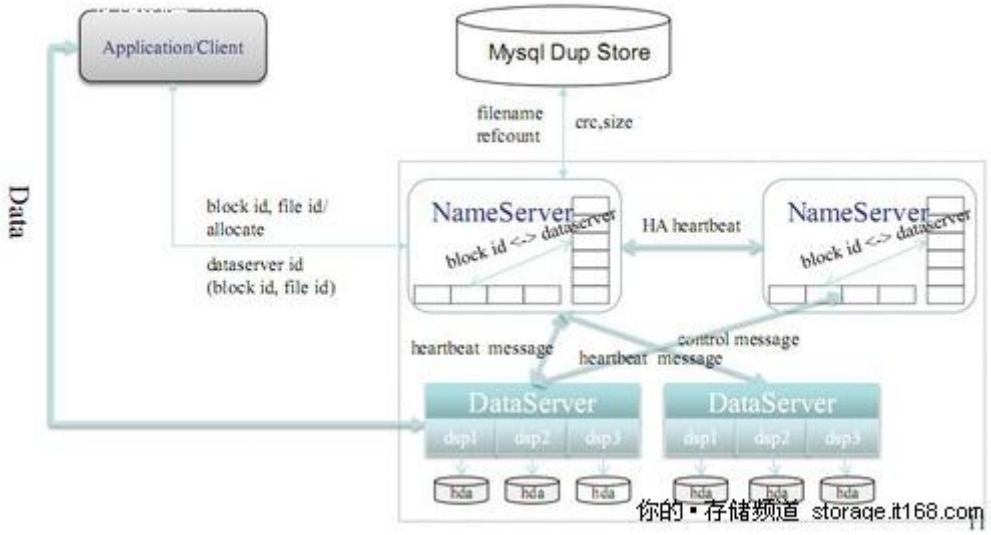
淘宝 TFS 文件系统在核心设计上最大的取巧的地方就在，传统的集群系统里面元数据只有 1 份，通常由管理节点来管理，因而很容易成为瓶颈。而对于淘宝网的用户来说，图片文件究竟用什么名字来保存实际上用户并不关心，因此 TFS 在设计规划上考虑在图片的保存文件名上暗藏了一些元数据信息，例如图片的大小、时间、访问频次等等信息，包括所在的逻辑块号。而在元数据上，实际上保存的信息很少，因此元数据结构非常简单。仅仅只需要一个 fileID，能够准确定位文件在什么地方。

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。

TFS 1.3 版本的集群文件系统

到 2009 年 6 月，TFS 1.3 版本上线，集群规模大大扩展，部署到淘宝的图片生产系统上，整个系统已经从原有 200 台 PC 服务器扩增至 440 台 PC Server(300G*12 SAS 15K RPM)+30 台 PC Server (600G*12 SAS 15K RPM)。支持文件数量也扩容至百亿级别；系统部署存储容量：1800TB(1.8PB)；当前实际存储容量：995TB；单台 Data Server 支持随机 IOPS 900+，流量 15MB+;目前 Name Server 运行的物理内存是 217MB(服务器使用千兆网卡)。



TFS 1.3 版本逻辑结构图

图为 TFS1.3 版本的逻辑结构图，在 TFS1.3 版本中，淘宝网的软件工作组重点改善了心跳和同步的性能，最新版本的心跳和同步在几秒钟之内就可完成切换，同时进行了一些新的优化：包括元数据存内存上，清理磁盘空间，性能上也做了优化，包括：

- 完全扁平化的数据组织结构，抛弃了传统文件系统的目录结构。
- 在块设备基础上建立自有的文件系统，减少 EXT3 等文件系统数据碎片带来的性能损耗。
- 单进程管理单块磁盘的方式，摒除 RAID5 机制。
- 带有 HA 机制的中央控制节点，在安全稳定和性能复杂度之间取得平衡。
- 尽量缩减元数据大小，将元数据全部加载入内存，提升访问速度。
- 跨机架和 IDC 的负载均衡和冗余安全策略。
- 完全平滑扩容。

在后面“图片服务器部署与缓存”一节中详细介绍了淘宝网整个图片处理系统的拓扑图。我们可以看到，TFS 在淘宝的部署环境中前端有两层缓冲，到达 TFS 系统的请求非常离散，所以 TFS 内部是没有任何数据的内存缓冲的，

包括传统文件系统的内存缓冲也不存在。

TFS 主要的性能参数不是 IO 吞吐量，而是单台 PCServer 提供随机读写 IOPS。由于大家硬件型号不同，当然也是因为一些技术保密的原因，淘宝网很难给出一个参考值来说明性能。但基本上可以达到单块磁盘随机 IOPS 理论最大值的 60%左右，整机的输出随盘数增加而线性增加。

开发中的 TFS2.0 与开源 TFS

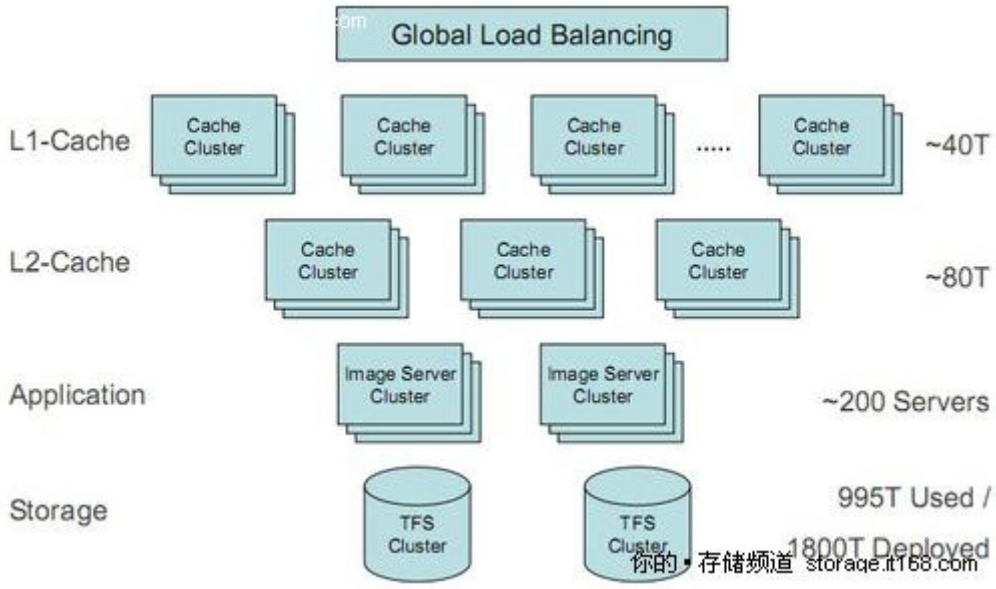
TFS 2.0 已经在开发过程中，主要解决的问题是大文件存储的难题。TFS 最早开发的时候针对小文件频繁并发读取的难题而开发，设计的块大小是 64MB，意味着每个文件小于 64MB，这对于一般的图片存储来说完全足够用了，但对于大文件存储还有一些瓶颈。

TFS 2.0 将重点针对大文件跨越块的存储进行优化。此外，还包括 SSD、SAS 硬盘不同硬盘特性的应用优化。根据淘宝网的资料数据，SSD 的存储成本大约是 20￥每 GB 左右，SAS 硬盘的存储成本约在 5-6￥每 GB，SATA 盘的每 GB 成本不到 1￥。随着对应用性能的要求提升，应用 SSD 是未来的趋势，针对不同硬盘的存取特性进行优化是十分必要的。

此外，章文嵩宣布，**TFS 将在 9 月份完全开源**，完全开源则意味着淘宝网将提供所有的源代码，开源版本的

TFS 将与淘宝网线上应用的系统完全一致。

图片服务器部署与缓存



淘宝网图片存储与处理系统全局拓扑，图片服务器前端还有一级和二级缓存服务器，尽量让图片在缓存中命中，最大程度的避免图片热点，实际上后端到达 TFS 的流量已经非常离散和平均

上图为淘宝网整体系统的拓扑图结构。整个系统就像一个庞大的服务器一样，有处理单元、缓存单元和存储单元。前面已经详细介绍过了后台的TFS集群文件存储系统，在 TFS 前端，还部署着 200 多台图片文件服务器，用 Apatch 实现，用于生成缩略图的运算。

这里需要补充一点，根据淘宝网的缩略图生成规则，缩略图都是实时生成的。这样做的好处有两点：一是为了

避免后端图片服务器上存储的图片数量过多，大大节约后台存储空间的需求，淘宝网计算，采用实时生成缩略图的模式比提前全部生成好缩略图的模式节约90%的存储空间，也就是说，存储空间只需要后一种模式的10%；二是，缩略图可根据需要实时生成出来，更为灵活。

图片文件服务器的前端则是一级缓存和二级缓存，前面还有全局负载均衡的设置，解决图片的访问热点问题。图片的访问热点一定存在，重要的是，让图片尽量在缓存中命中。目前淘宝网在各个运营商的中心点设有二级缓存，整体系统中心店设有一级缓存，加上全局负载均衡，传递到后端TFS的流量就已经非常均衡和分散了，对前端的响应性能也大大提高。

根据淘宝的缓存策略，大部分图片都尽量在缓存中命中，如果缓存中无法命中，则会在本地服务器上查找是否存有原图，并根据原图生成缩略图，如果都没有命中，则会考虑去后台TFS集群文件存储系统上调取，因此，最终反馈到TFS集群文件存储系统上的流量已经被大大优化了。

淘宝网将图片处理与缓存编写成基于Nginx的模块，淘宝网认为Nginx是目前性能最高的HTTP服务器(用户空间)，代码清晰，模块化非常好。淘宝网使用GraphicsMagick进行图片处理，采用了面向小对象的缓存文件系统，前端有 LVS+Haproxy 将原图和其所有缩略图请求都调度到同

一台 Image Server。

文件定位上，内存用 hash 算法做索引，最多一次读盘。写盘方式则采用 Append 方式写，并采用了淘汰策略 FIFO，主要考虑降低硬盘的写操作，没有必要进一步提高 Cache 命中率，因为 Image Server 和 TFS 在同一个数据中心，读盘效率还是非常高的。