

# 稳定、高效、低碳

## --淘宝软件基础设施构建实践

章文嵩（正明）

淘宝核心系统

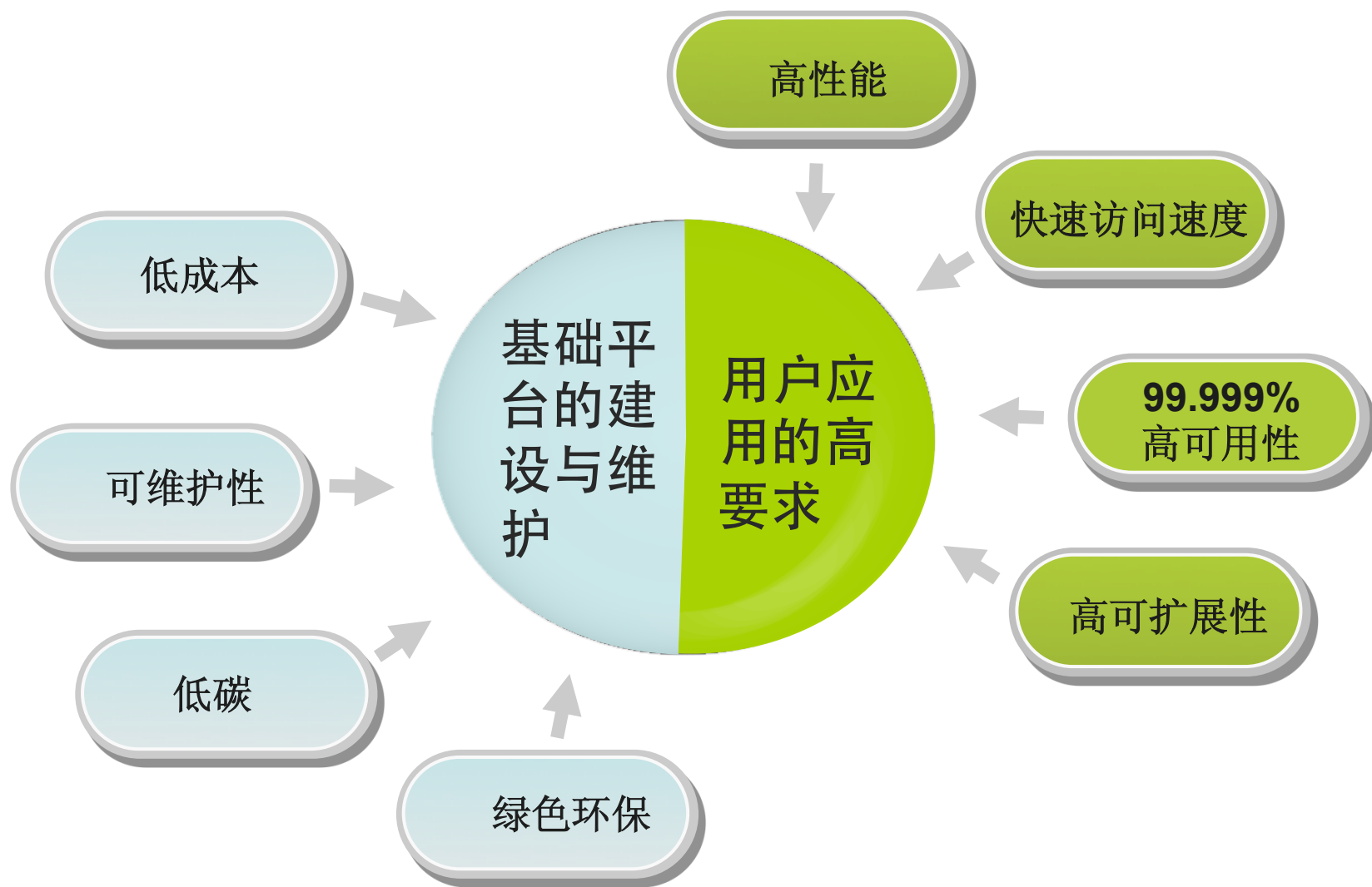
2010.12.7

Velocity China 2010



- 一、基础设施的规模与挑战
- 二、事例：图片存储与CDN系统
- 三、软件基础设施项目
- 四、淘宝开源平台
- 五、小结

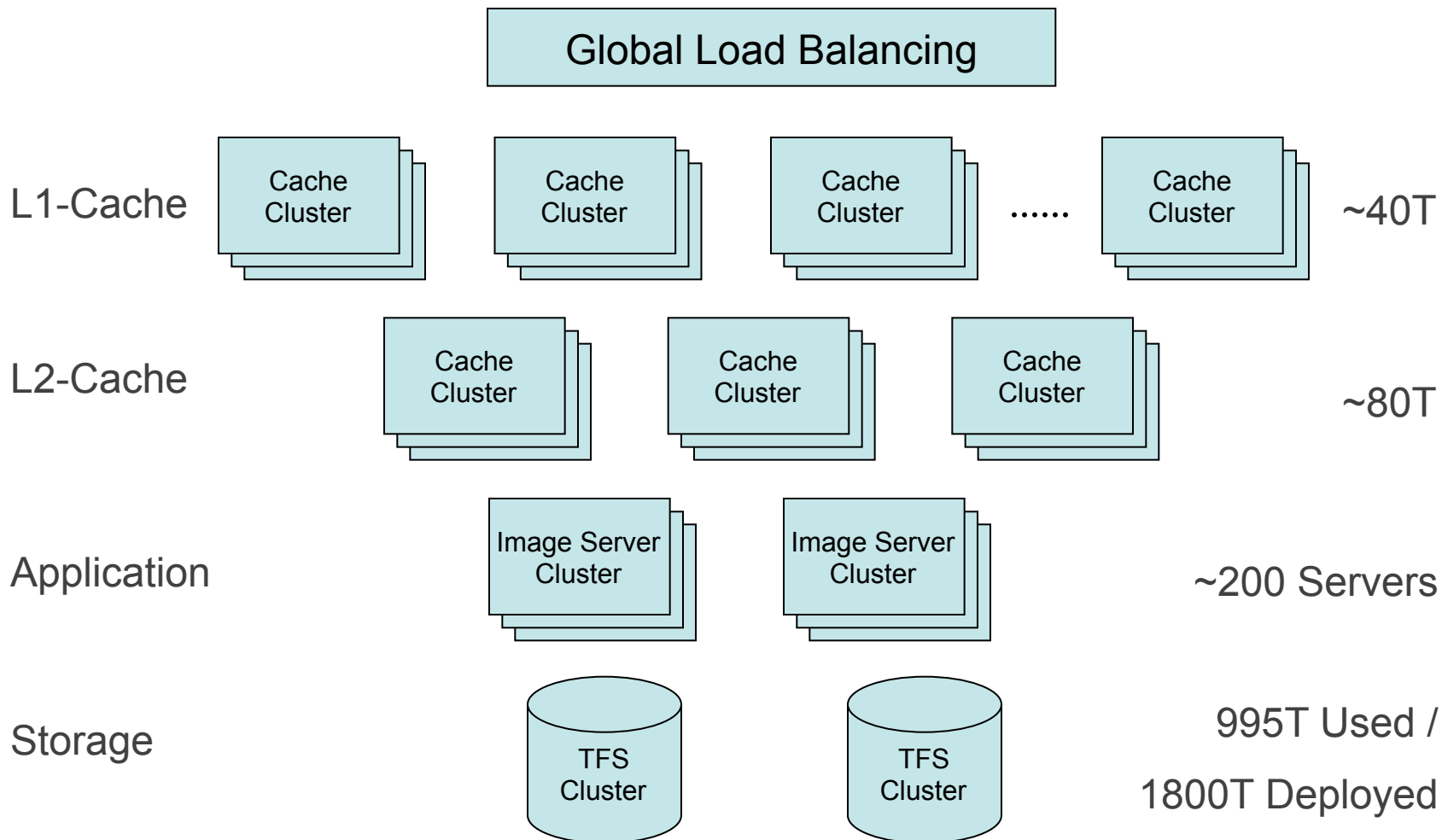
- Alexa全球排名为13，生产服务器约为15000台
- 系统需要足够的容量随时应对突发的流量增长
- 淘宝的下一个十年：大淘宝
  - 提供电子商务基础设施服务
  - 打造电子商务生态系统
- 增长
  - 业务每年以超过100%的速度增长
  - 可预见未来十年有百倍以上规模变化
- 技术是电子商务基础设施服务的核心竞争力



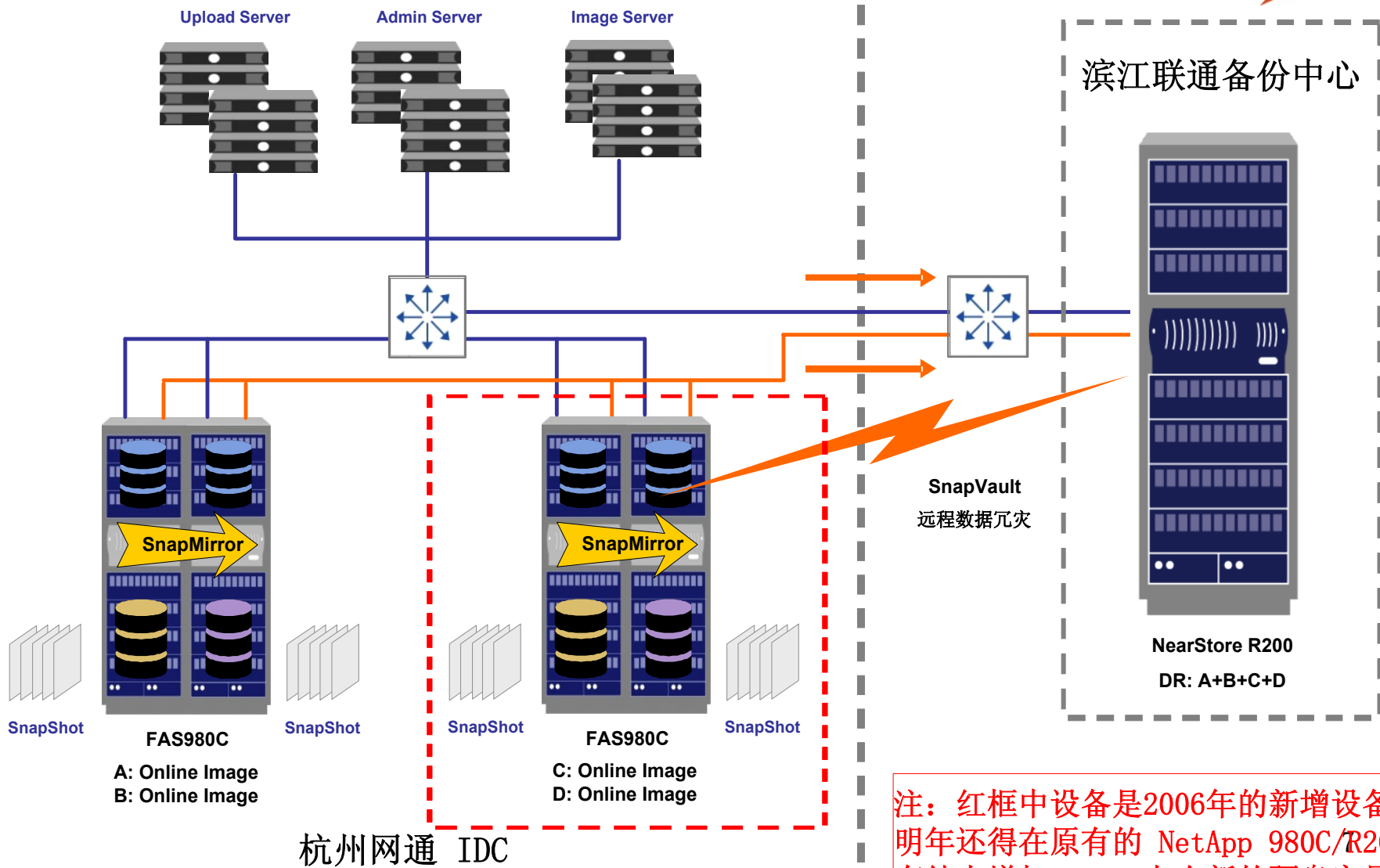


- 一、基础设施的规模与挑战
- 二、事例：图片存储与CDN系统
- 三、软件基础设施项目
- 四、淘宝开源平台
- 五、小结

# 图片存储与CDN系统全貌



# 2007年之前的图片存储系统



注：红框中设备是2006年的新增设备，明年还得在原有的 NetApp 980C/R200 存储上增加 20TB 左右新的硬盘容量。

- 系统需求
  - 淘宝的影响越来越大，数据的安全也更加重要
  - 数据存储量以每年二倍的速度增长（即原来的三倍）
- 商用存储产品
  - 对小文件的存储无法优化
  - 文件数量大，网络存储设备无法支撑
  - 连接的服务器越来越多，网络连接数已经到达了网络存储设备的极限
  - 扩容成本高，10T的存储容量需要几百万¥
  - 单点，容灾和安全性无法得到很好的保证



- 2007年6月

淘宝自主开发的分布式的文件系统

TFS (Taobao File System) 1.0上线运行

主要解决海量小文件的分布式存储

集群规模: 200台PC Server(146G\*6 SAS 15K Raid5)

文件数量: 亿级别

系统部署存储容量: 140 TB

实际使用存储容量: 50 TB

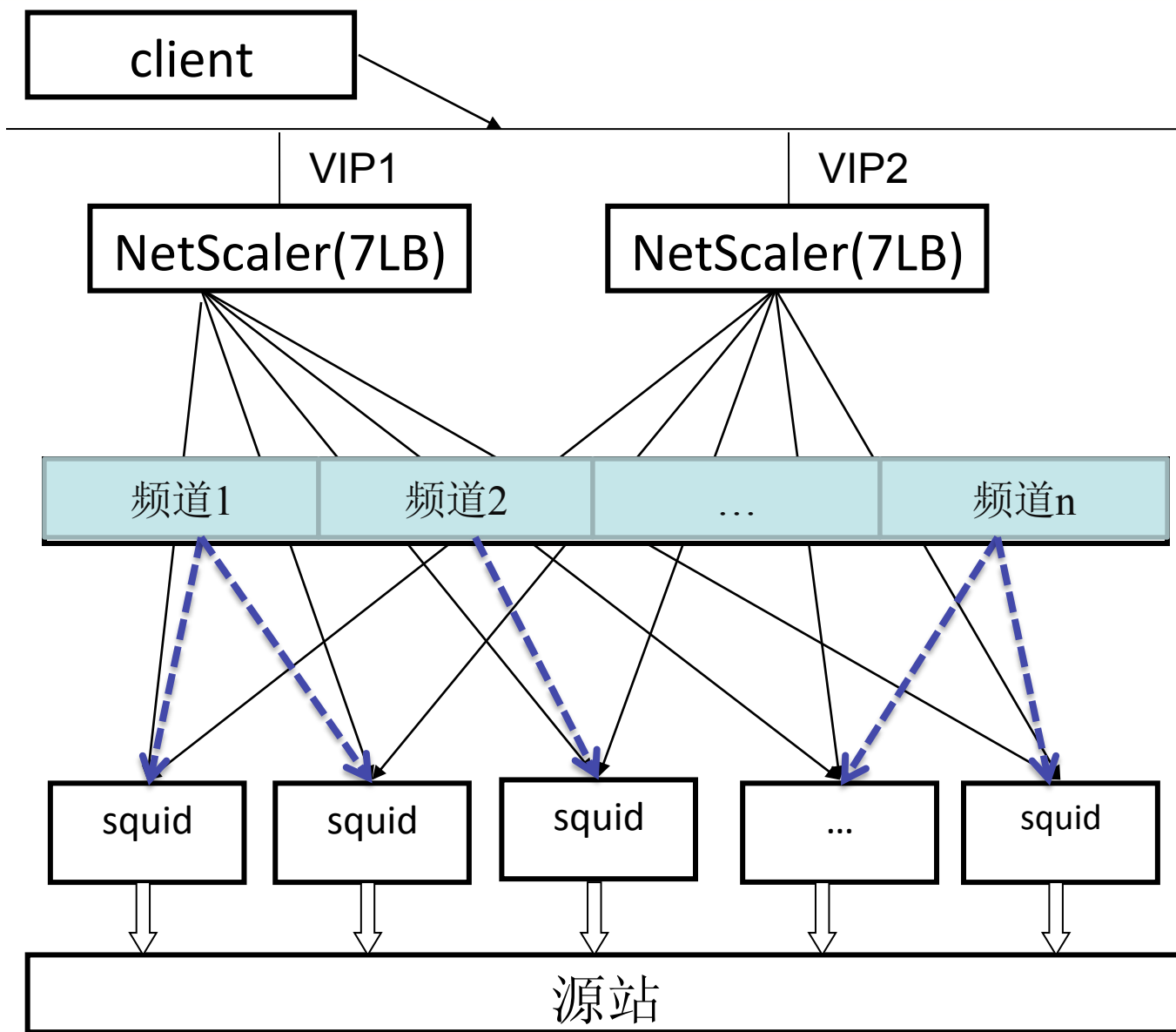
单台支持随机IOPS 200+, 流量3MBps

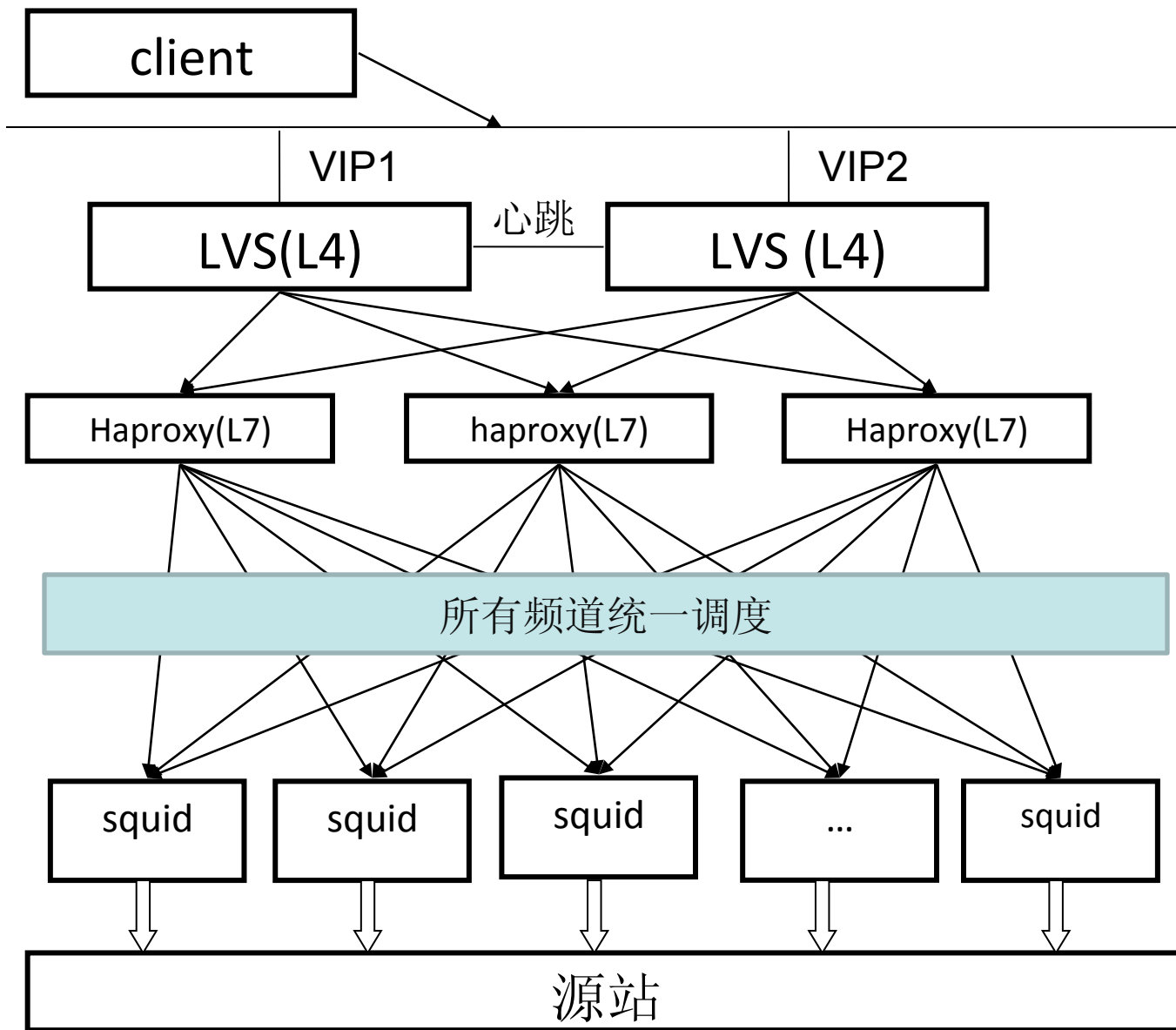
- 集群由一对Name Server和多台Data Server构成
- 每个Data Server运行在一台普通的Linux主机上
- 以block文件的形式存放数据文件(一般64M一个block)
- block存多份保证数据安全
- 利用ext3文件系统存放数据文件
- 磁盘raid5做数据冗余
- 文件名内置元数据信息，用户自己保存TFS文件名与实际文件的对照关系 - 使得元数据量特别小

- 2009年6月  
TFS (Taobao File System) 1.3上线运行
- 集群规模 (2010.8.22)
  - 440台PC Server (300G\*12 SAS 15K RPM) + 30台PC Server (600G\*12 SAS 15K RPM)
  - 文件数量: 百亿级别
  - 系统部署存储容量: 1800 TB
  - 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)

- TFS1.3提供了一些重要的功能特性
  - 所有的元数据全部都内存化
  - 清理磁盘空洞
  - 容量和负载的均衡策略
  - 平滑的扩容
  - 数据安全性的冗余保证
  - 几秒内完成Name Server故障自动切换
  - 容灾策略
  - 性能大幅提升

- CDN服务的图片规模
  - 约250T容量的原图 + 250T容量的缩略图
  - 约286亿左右的图片数，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%
- CDN部署规模
  - 41个节点，部署在网民相当密集的中心城市
  - 每个节点目前处理能力在10G或以上
  - CDN部署的总处理能力已到410G以上
  - 目前承载淘宝流量高峰时318G流量





对比项 \ 节点	新架构	老架构
流量分布均匀性	☆☆☆☆☆	☆☆☆
可维护性	☆☆☆	☆☆☆
抗攻击能力	☆☆☆☆	☆☆☆☆
自主控制能力	☆☆☆☆☆	☆☆☆
价格	☆☆☆☆☆	☆☆☆
扩展能力	☆☆☆☆☆	☆☆
灵活性	☆☆☆☆☆	☆☆

- 流量分布均匀性：所有的频道统一调度到128台squid，而不是将squid按频道分组，可提高命中率2%以上
- 扩展能力：在一个VIP上新架构可以扩展到近100G的流量（当然要用万兆网卡）
- 灵活性：一致性Hash调度方法使得增加和删除服务器非常方便，只有 $1/(n+1)$ 的对象需要迁移



- 在COSS存储系统基础上实现了TCOSS，FIFO加上按一定比例保留热点对象，支持1T大小的文件
- Squid内存优化，一台Squid服务器若有一千万对象，大约节省1250M内存，更多的内存可以用作memory cache
- 用sendfile来发送缓存在硬盘上的对象，加上page cache，充分利用操作系统的特性
- 针对SSD硬盘，可以采用DIRECT\_IO方式访问，将内存省给SAS/SATA硬盘做page cache
- 在Squid服务器上使用SSD+SAS+SATA混合存储，实现了类似GDSF算法，图片随着热点变化而迁移

$$migration\_weight * \frac{frequency}{size^{migration\_power}} ; migration\_power \in (0, 1]$$

- 简单按对象大小划分：小的进SSD，中的放SAS，大的存SATA
- SSD + 4 \* SAS + SATA上的访问负载如下：

```
[root@cache161 ~]# iostat -x -k 60 | egrep -v -e "sd.[1-9]"
```

```
...
```

```
avg-cpu:  %user  %nice %system %iowait  %steal   %idle
           3.15   0.00   5.63  11.35   0.00  79.87
```

Device:	rrqm/s	wrqm/s	r/s	w/s	rkB/s	wkB/s	avgrq-sz	avgqu-sz	await	svctm	%util
sda	15.40	1.17	50.66	2.63	2673.22	124.85	105.01	0.55	10.39	6.27	33.41
sdb	0.07	0.03	447.29	1.02	4359.01	191.90	20.30	0.32	0.71	0.27	12.13
sdc	5.73	1.53	114.93	8.42	1264.86	100.58	22.14	1.05	8.48	3.56	43.94
sdd	5.57	2.07	121.83	9.57	1319.45	104.12	21.67	1.19	9.02	3.63	47.72
sde	5.53	1.45	111.45	8.52	1246.53	101.92	22.48	0.95	7.88	3.42	41.06
sdf	5.45	2.02	118.93	8.00	1281.92	106.25	21.87	1.19	9.37	3.74	47.44

其中：黑色为SATA，绿色为SSD，红色为SAS  
4块SAS硬盘上的访问量和超过SSD硬盘上的访问量

- 按对象访问热点进行迁移：最热的进SSD，中等热度的放SAS，轻热度的存SATA
- SSD + 4 \* SAS + SATA上的访问负载如下：

```
[root@cache161 ~]# iostat -x -k 60 | egrep -v -e "sd.[1-9]"
```

```
...
```

```
avg-cpu:  %user  %nice %system %iowait  %steal   %idle
           3.15   0.00   5.63  11.35   0.00  79.87
```

Device:	rrqm/s	wrqm/s	r/s	w/s	rkB/s	wkB/s	avgrq-sz	avgqu-sz	await	svctm	%util
sda	5.08	1.65	18.55	2.52	1210.07	119.00	126.18	0.14	6.50	5.46	11.51
sdb	1.68	0.05	610.53	1.75	6962.29	413.47	24.09	0.28	0.46	0.23	14.25
sdc	0.22	0.03	28.87	0.97	1172.93	189.13	91.31	0.16	5.28	4.40	13.13
sdd	0.23	0.02	29.70	0.77	1133.47	122.53	82.45	0.15	4.99	4.39	13.37
sde	0.18	0.03	28.23	1.03	1078.73	206.27	87.81	0.15	5.00	4.24	12.40
sdf	0.10	0.02	28.42	0.55	1090.27	115.00	83.22	0.15	5.04	4.44	12.86

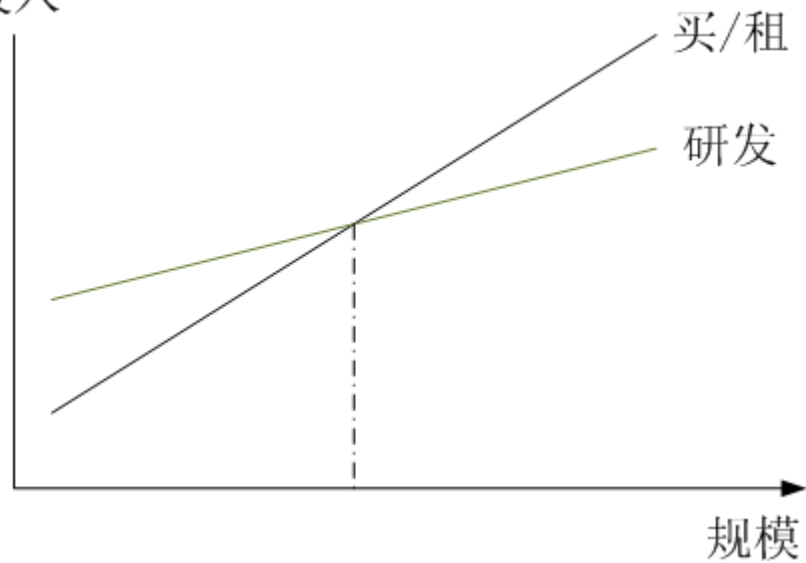
其中：黑色为SATA，绿色为SSD，红色为SAS

SSD硬盘上的访问量是4块SAS硬盘上访问量之和的5倍以上，SAS和SATA的硬盘利用率低了很多

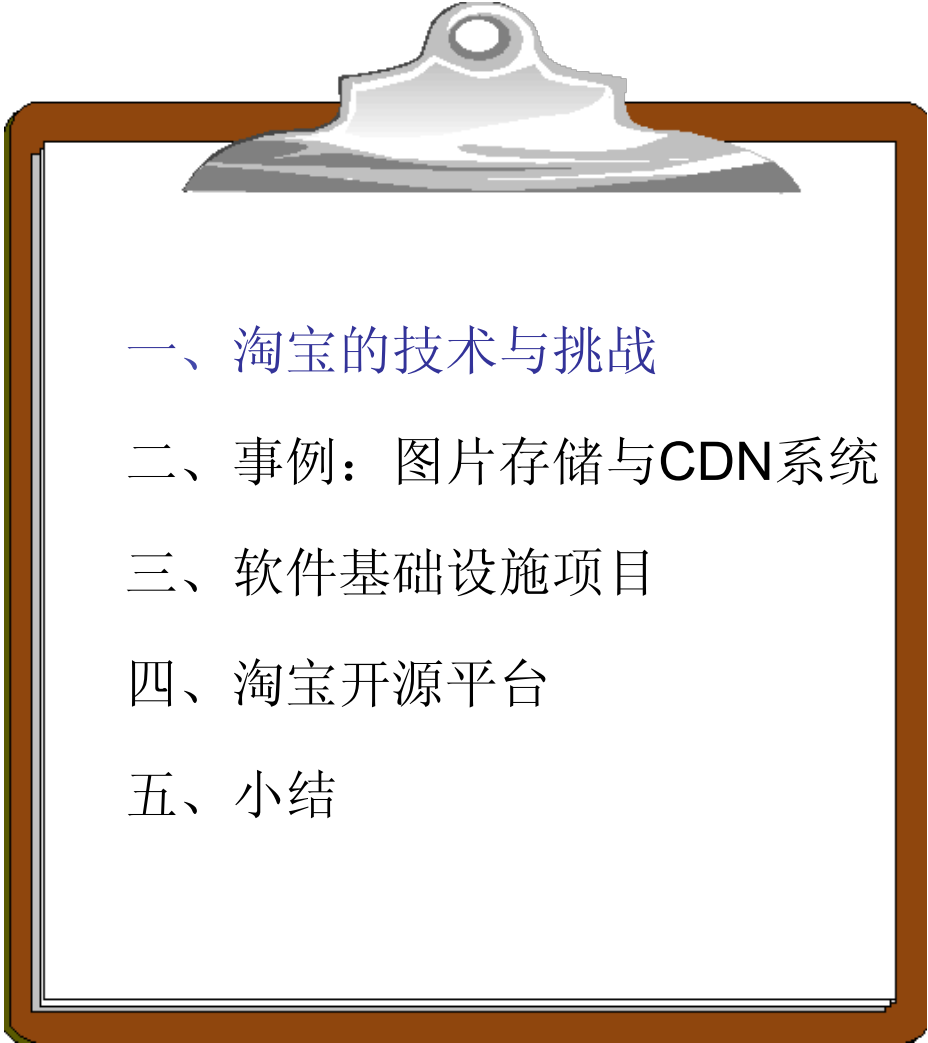
- 节点规模：32台 DELL R710服务器
- 逻辑结构：2 LVS + 32 Haproxy + 64 Squid
- 时间：12月21日上线运行
- 当前最大服务流量：10.58 Gbps
- 理论最大负载能力：15Gbps以上
- 单台R710服务器可到500Mbps以上的吞吐率
- 单squid最大object数目：1800万
- Cache请求命中率：97%
- Cache字节命中率：97%
- **最重要的是命中率提高，大大改善用户的访问体验**

- 节点规模：30台 DELL PowerEdge 2950服务器
- 逻辑结构：2 LVS + 30 Haproxy + 60 Squid
- 时间：2010年5月上线运行
- 理论最大负载能力：12Gbps
- 单台2950服务器可到400Mbps的吞吐率
- 单台存储：160G SSD + 143G SAS \* 4 + 1T SATA
- 单squid最大object数目：3000万
- Cache请求命中率：97.5%
- Cache字节命中率：97.5%
- **最重要的是命中率提高，大大改善用户的访问体验**

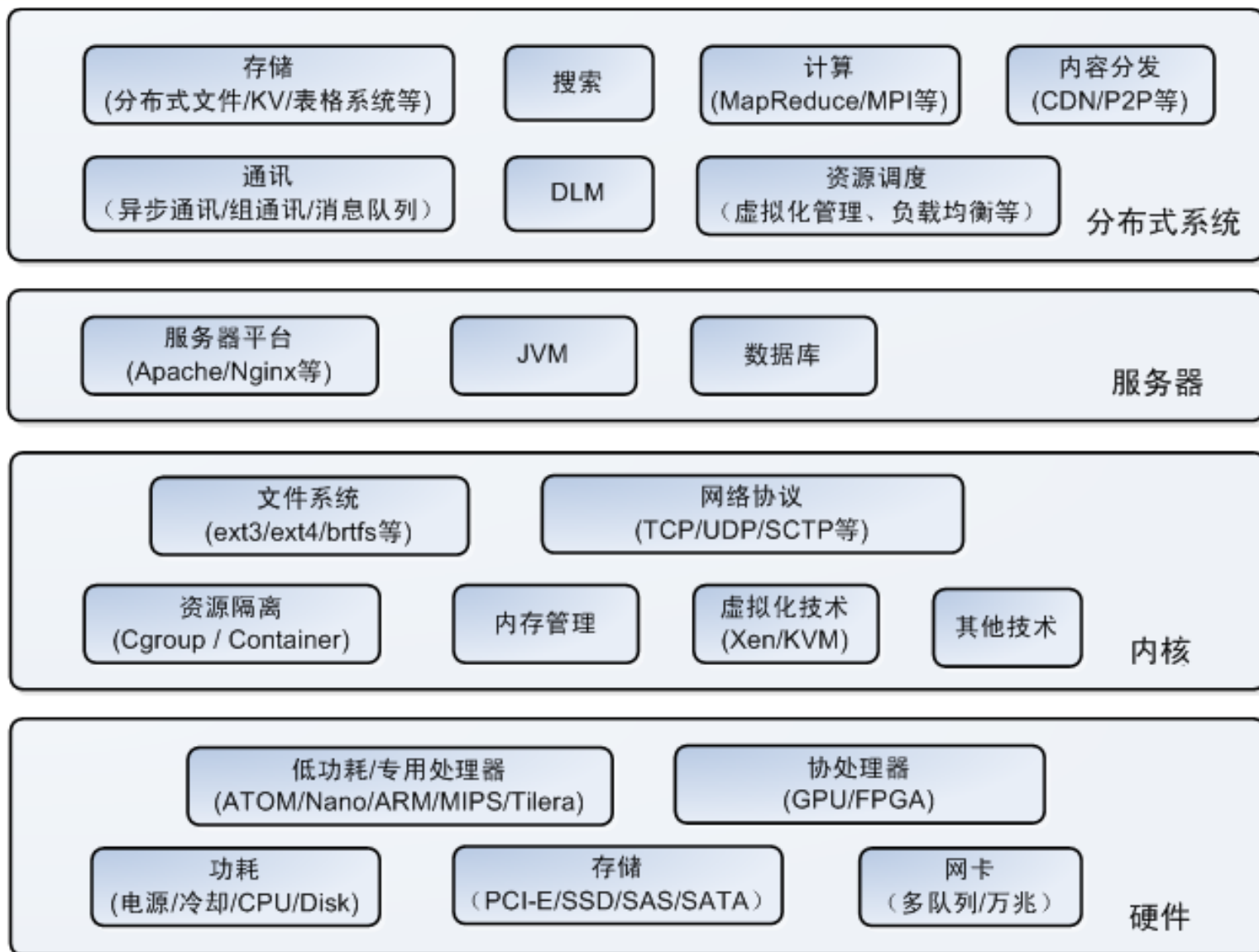
- 商用软件不能满足大规模系统的需求
- 采用开源软件与自主开发相结合，有更好的可控性和更大的优化空间，系统上有更高的可扩展性
- 规模效应，研发投入者投入



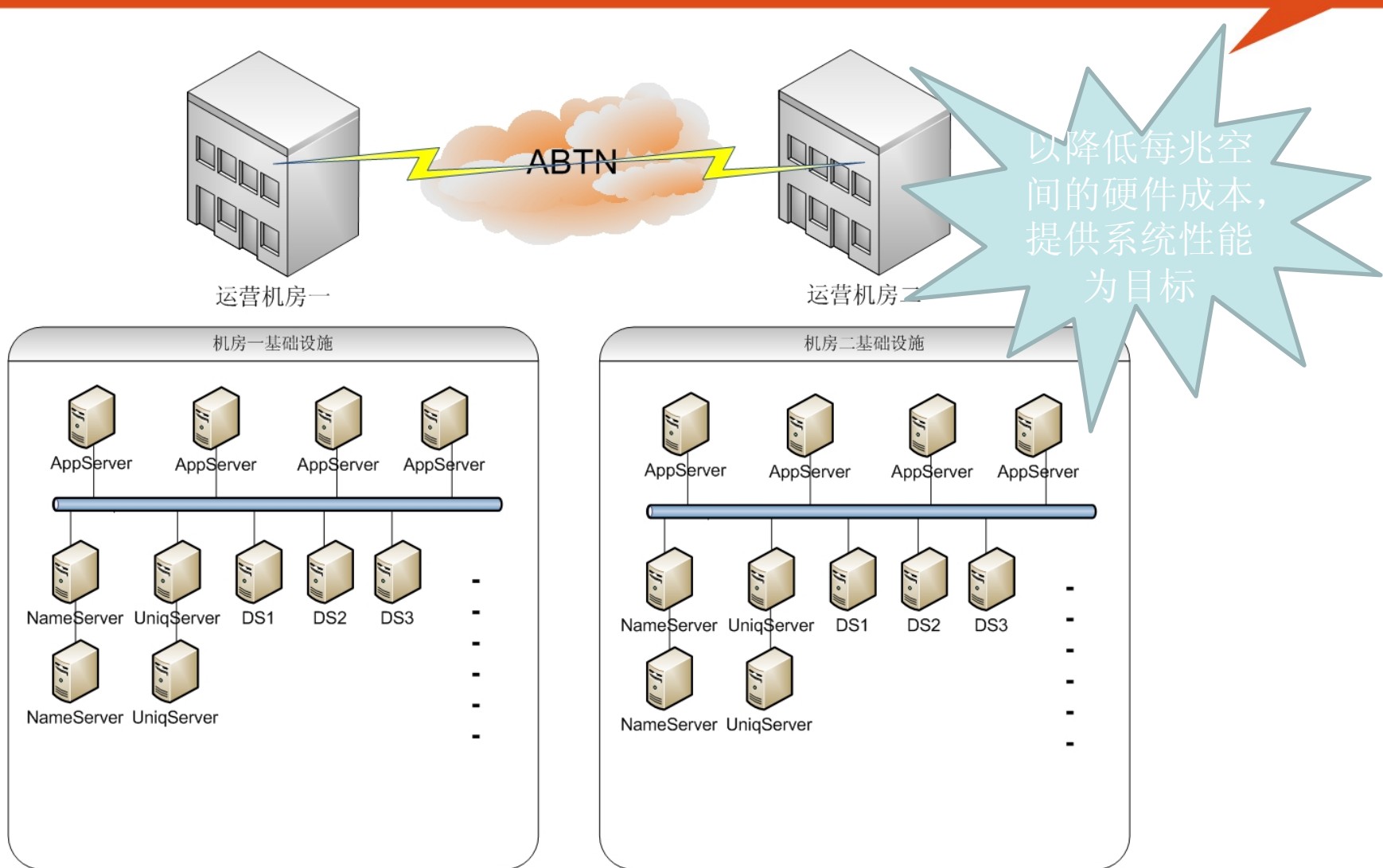


- 
- 一、淘宝的技术与挑战
  - 二、事例：图片存储与CDN系统
  - 三、软件基础设施项目
  - 四、淘宝开源平台
  - 五、小结

# 软件基础设施的规划







- **TFS系统的主要特点**

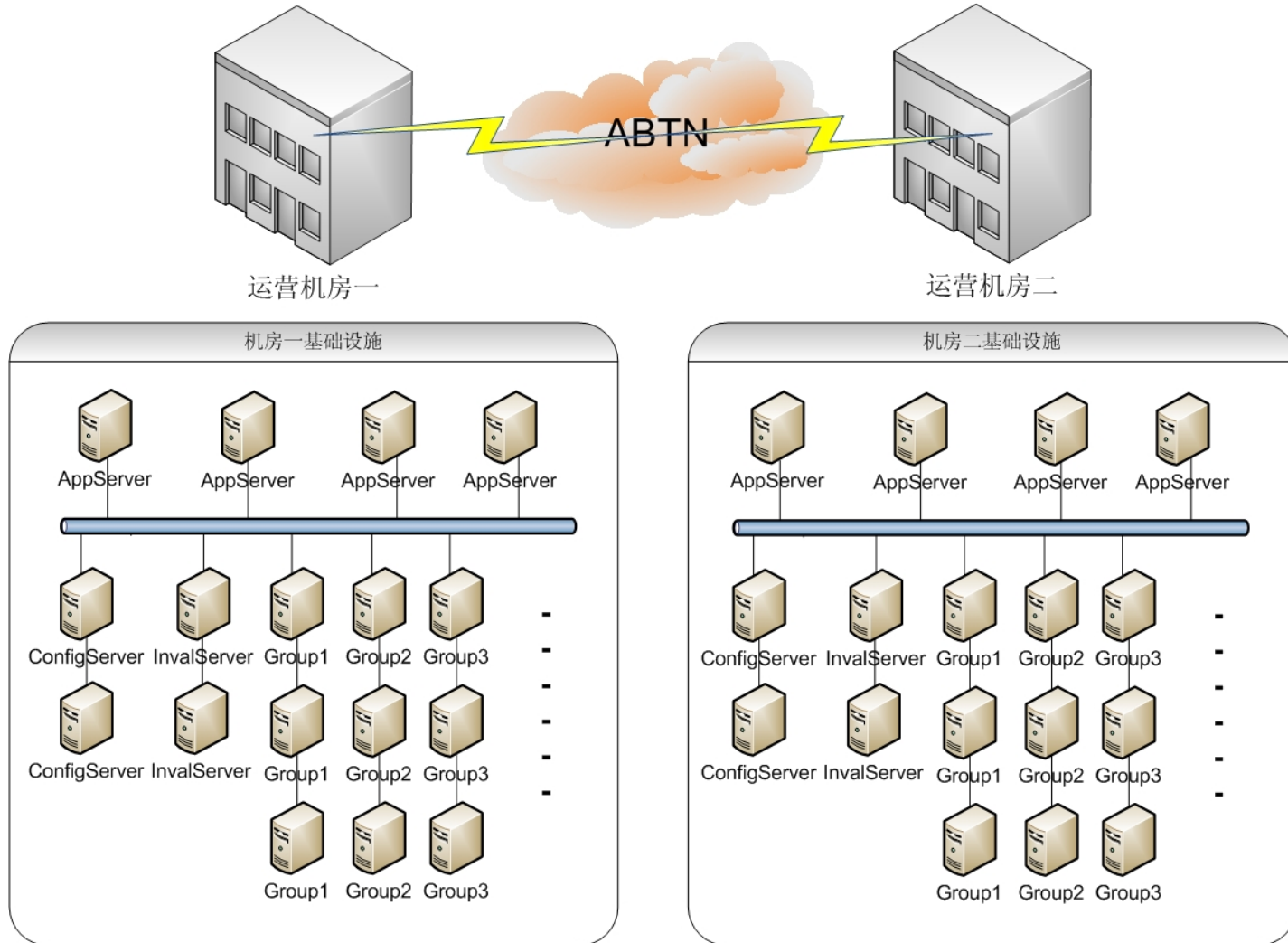
- 针对互联网应用，文件命名不敏感
- 架构当前主要解决一致性和可分区性
- 提供双机房备份，解决数据容灾
- 多进程分别管理单独的磁盘，充分利用系统资源
- 自定义文件系统+支持EXT3/EXT4，利用操作系统的特性

- **TFS系统在淘宝的主要应用**

- 支持商品非结构化及交易中间数据存储
- 数百台服务器，2P存储容量，百亿单位文件数量

- **TFS系统未来的发展**

- 自描述的大文件支持
- 部分支持自命名文件
- 持续提升单机性能及集群扩充能力，降低成本



- **TAIR系统的主要特点**

- DHT数据分布，专注高性能KV类型数据访问
- 架构当前主要解决可用性和可分区性
- 轻量级中心控制节点，简化架构设计
- 提供双机房备份，解决数据容灾
- Json形式提供部分结构化数据访问
- 支持MEM/BDB/EXT3/EXT4，提供不同性能及安全要求的数据服务

- **TAIR系统在淘宝的主要应用**

- 作为缓存支持淘宝绝大部分应用，持久化引擎当前支持部分应用
- 百台服务器，数百T存储容量，数百亿访问/天

- **TAIR系统未来的发展**

- 灵活的配置策略，应用自定义CAP平衡
- 持续提升单机性能及集群扩充能力，降低成本

- 淘宝数据库系统有不少已经接近百亿条的记录
  - 历史交易记录有40多亿条
  - 用户收藏夹记录超过60亿条，等等
  - 目前不断地拆分数数据库，不是长远的办法
- OceanBase是海量信息存储与检索的线上系统
  - 解决千亿条级别的表格系统
  - 有严格的响应时间要求，平均低于1~10ms
  - 系统的吞吐率要能达到1200QPS/每台
  - 预计2011Q1完成第一个应用的线上部署
  - 以后针对SSD设备做优化

- 提高MySQL数据库系统的整体性能
  - 数据增量复制，用于数据挖掘和容灾
  - 主从服务器并行同步
  - 改善日志系统，提高集群的容灾能力
- 提升MySQL单机性能
  - 利用PCI-E Flash卡做存储cache，大幅提升性能和单表容量
  - 优化和改进Innodb引擎，提高锁和IO的效率
- MySQL负载均衡解决方案
  - TDDL, JDBC层面
  - 轻量代理服务器，对客户端完全透明

- 适合淘宝的高性能服务器开发
  - Apache/Nginx等服务器软件的定制、优化和管理
  - 开发与维护淘宝公用基础HTTP模块
  - Squid的定制与优化
  - 安全策略服务器
- 大规模负载均衡解决方案
  - LVS的开发与维护
  - HAProxy的定制与开发
- 网络协议栈调优
  - TCP协议优化、内核参数调优

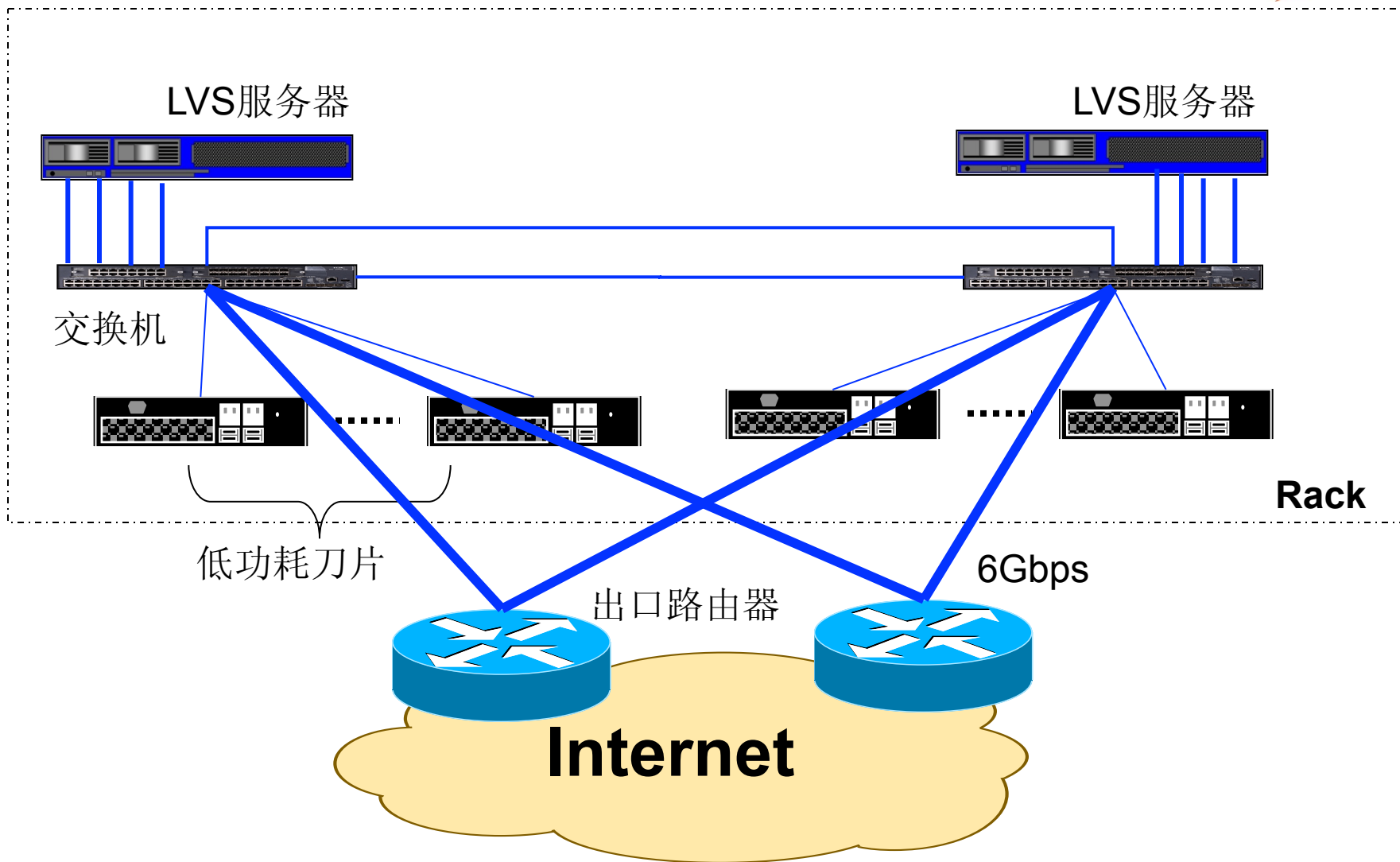
- 从多个层次为Java应用提供高效的底层平台
  - 高效的Java基础库
  - Java虚拟机优化，针对Hotspots VM及Open JDK VM
  - Linux操作系统层优化
  - Hypervisor, Java without OS
  - 专用硬件
- 策略：容易的先做
- 目前进展
  - 针对特定CPU，对JVM实现所依赖的相关库进行汇编重写
  - 针对Intel CPU对JVM进行编译级优化
  - 修改JVM，实现GC Invisible Heap技术



- 针对淘宝的应用负载来维护一个稳定、高效的Linux内核
  - 目前涉及文件系统、资源管理、网络等子系统
  - 与其他团队合作，针对业务系统优化性能和资源分配
  - 根据服务器的不同硬件配置定制Linux内核
- 同Linux内核开发社区合作
  - 在开发、测试、缺陷修复等各方面与社区全方位合作
  - 包括新特性和缺陷修复等工作成果都将反馈到社区

- 定位：基于代理模型的组通讯中间件，淘宝和阿里巴巴集团研究院联合开发
- 应用现状
  - 稳定应用在淘宝广告引擎系统，广告系统中的脊椎
  - 阿里云飞天系统中的核心组件，近千台服务器的机群规模
- 后续发展
  - 单集群数千台规模的通信能力
  - 持续提升通信性能

- 低功耗硬件平台
  - 低功耗的CPU，如Intel ATOM, VIA Nano等
  - 低功耗的Chipset; SSD或低功耗的SATA硬盘
  - 关闭GPU和USB Controller等
- 适用不需要太多CPU计算的I/O类型应用
  - 例如CDN Cache Server、memory cache、存储节点、静态文件Web Server等
- 好处（大大降低成本）：
  - 降低电力消耗，减少碳排放
  - 单位空间(机柜)下有更高的I/O吞吐率
  - 降低硬件购置成本和运营成本



- 主要解决现有的问题
  - 商用产品的性能瓶颈、功能欠缺，以及不稳定性
  - 整个系统的规模、性能、可用性、成本和可管理性
- 开发完全自主的CDN系统
  - CDN节点的新架构和优化
  - CDN监控平台
  - 全局流量调度系统支持基于节点负载状态调度和基于链路状态调度
  - CDN实时图片删除
  - CDN访问日志过滤系统
  - 配置管理平台

- CDN系统的研发与运维
  - 针对教育网的CDN部署进一步优化方案
  - 动态页面加速，节点间应用级路由
  - 持续提高节点性能（应用软件、操作系统等）
  - 优化GTM全局调度系统
  - 持续提高CDN系统可运维性，完善CDN内容管理系统
- CDN系统的建设
  - 思路正在转向“部署更多的小节点，尽可能离用户近一些”
  - 定制化和快速部署



- 一、基础设施的规模与挑战
- 二、事例：图片存储与CDN系统
- 三、软件基础设施项目
- 四、淘宝开源平台
- 五、小结

Home | Taobao code open Source - Mozilla Firefox

文件(F) 编辑(E) 查看(V) 历史(S) 书签(B) 工具(T) 帮助(H)

http://code.taobao.org/

Home | Taobao code open Source

taobao code » creative pool » project » welcome,zhengming » publish creative » create project » logout »

## TaobaoCODE

creative   Hot tag: test nginx php java C# cms utwei 敏捷管理 Agile

### Manage

- my Creatives
- my Projects

### creative category(0) more

### project category(4) more

- system (1)
- Storage (2)
- utils (1)
- development (2)

### Help more

### hot project

[nginx\\_concat\\_module](#)  
This is an Nginx module, which can be used to concatenate multiple files into a single file. E.g. by concatenating/combining CSS and JavaScript files, you can minimize the number of HTTP requests then speed up your website. Basically, it's an Nginx version of Apache mod\_concat, but with more features.  
shudu create at 2010-06-30 18:26:48 [view](#) » [comment](#) » [wiki](#) » [join project](#) »

[AutoMan](#)  
AutoMan是淘宝测试team自主研发的包括自动化脚本编写框架,脚本执行,定时执行计划,报表分析,云测试环境管理等一体化自动化云测试平台。  
宝驹 create at 2010-06-30 15:02:25 [view](#) » [comment](#) » [wiki](#) » [join project](#) »

[tair-client-java](#)  
java client for tair  
ruohai create at 2010-06-29 18:06:26 [view](#) » [comment](#) » [wiki](#) » [join project](#) »

[tb-common-utils](#)  
taobao system library and network library  
MaoQi create at 2010-06-29 10:50:53 [view](#) » [comment](#) » [wiki](#) » [join project](#) »

[tair](#)  
Tair is a distributed, high performance key/value storage system  
ruohai create at 2010-06-29 09:41:05 [view](#) » [comment](#) » [wiki](#) » [join project](#) »

[taobaocode](#)  
旺旺群号:59353002  
残剑 create at 2010-06-23 19:16:39 [view](#) » [comment](#) » [wiki](#) » [join project](#) »

### Latest Creatives more

### Latest Projects more

- [nginx\\_concat\\_module](#)  
shudu 发布于2010-06-30 18:26:48
- [AutoMan](#)  
宝驹 发布于2010-06-30 15:02:25
- [tair-client-java](#)  
ruohai 发布于2010-06-29 18:06:26
- [tb-common-utils](#)  
MaoQi 发布于2010-06-29 10:50:53
- [tair](#)  
ruohai 发布于2010-06-29 09:41:05

### Activest Users

- [残剑](#)
- [路奇](#)
- [heiyeluren](#)
- [Will](#)
- [tufengwei](#)
- [迦勒](#)
- [ruohai](#)
- [shudu](#)

Contact Us - About

## 淘宝开源

完成

PageRank Alexa



- Code.taobao.org是开放的开源平台，淘宝公司在上面发布开源项目，也非常欢迎外部人员在上面发布开源项目
- 平台本身的管理软件也是一个开源项目
- 目前淘宝开源的项目：
  - 淘宝的Key/Value Cache/Store – TAIR
  - 淘宝分布式文件系统TFS
  - TaobaoABS等等
- 会陆续将淘宝的基础软件开源
- 淘宝希望以更开放的方式与业界一起进行技术创新



- 一、基础设施的规模与挑战
- 二、事例：图片存储与CDN系统
- 三、软件基础设施项目
- 四、淘宝开源平台
- 五、小结

- 采用开源软件与自主开发相结合，是构建大规模基础设施平台的必由之路
- 我们对系统的认识都是不断提高的
- 可以在软件和硬件多个层次优化
- 优化是长期持续的过程
- 淘宝--从使用技术走向创造技术的公司

Q & A  
谢谢!♪