

快一点，再快一点！

多层缓存系统在唱吧的实践经验

唱吧 田然



唱吧

怎么做一个够爽的App？

- 时间碎片化
 - 更高的速度要求
- 网络不稳定
 - 更差的网络条件
 - 甚至没有网络的情况下也要能访问
- 网络流量要求严格

天下武功，唯快不破



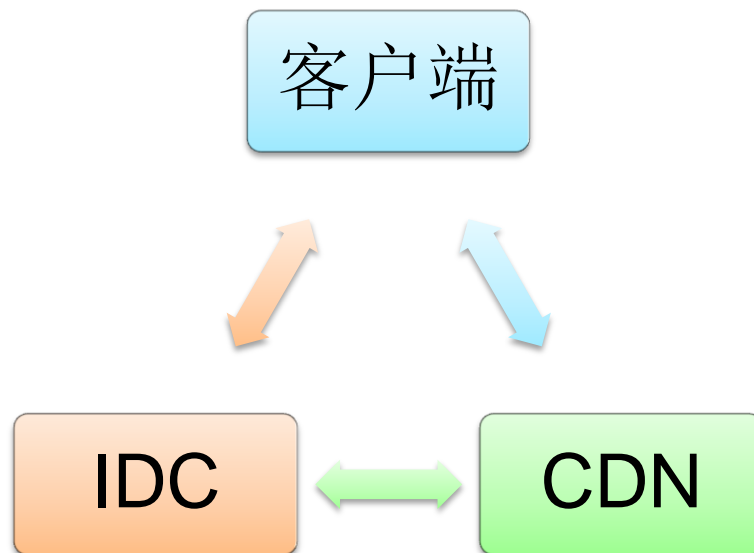
缓存系统

- Cache
 - 与其叫缓存，不如叫快取（信、达、雅）
- 一种实现方式，不是硬件设备



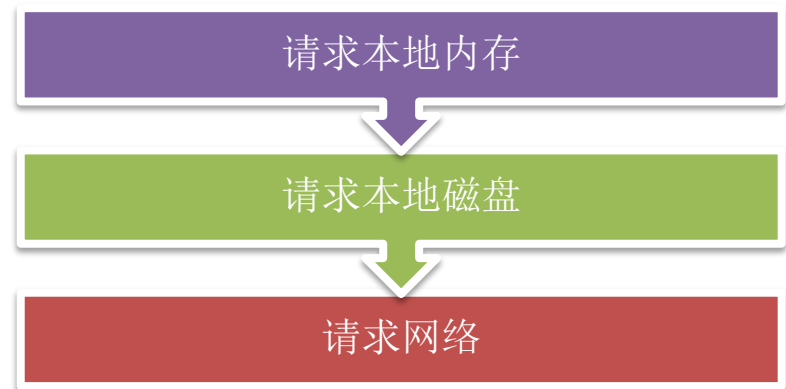
App时代的缓存

- 客户端缓存
 - 内存缓存
 - 文件缓存
 - 网络请求缓存
- 服务器端缓存
 - Memcache
 - Redis
 - 跨数据中心
- CDN
 - 选择、优化
 - 上云？



客户端缓存（1）

- 内存缓存
 - iOS
 - NSCache
 - Android
 - LRU Cache 限制可用内存比例
 - 特殊机型需要定制策略，例如更改图片分辨率，限制缓存页面层数
- 文件缓存
 - 网络下载过的文本、图片、音频等
 - 加快加载速度、减少网络流量



客户端缓存（2）

- 网络请求缓存
 - <http://api.changba.com/ktv.php?action=gethottestuserwork>
 - <md5(uri), serialized result> (json/xml)
 - <type, md5(uri), serialized result>
 - <type, md5(uri), serialized result, 30s>
 - <type, md5(uri), serialized result, 60s, 7day>
- 针对应用特征定制不同级别不同缓存空间的策略

	占用内存	占用磁盘	过期时间
App页面	较大	无	App生命周期
图片	大	小	长（天为周期）
音频	无	大	长（不过期）
网络请求	很小	小	特殊，通常较短

- 在客户端预留各个级别的缓存清理接口



服务器端缓存（1）

1000万安装量

100万日活跃用户

5万峰值同时在线

每秒5000次网络请求

- 2000次请求全局数据
- 2000次请求一般局部数据
- 1000次其他请求

服务器端缓存（2）

罗列一堆数字

- 内存寻址 0.1微秒
- 压缩1KB数据 10微秒
- **千兆网内发送2Kb的数据 20微秒**
- 内存中顺序读1 MB 0.25毫秒
- **Disk seek 8毫秒 ~ 12毫秒**
- 从网络顺序读1MB数据 10毫秒

以下为不负责任的数字

- Mysql每秒能处理的普通query数 > 1k
- Nginx 每秒能处理的普通query数 > 50k
- Memcache每秒能处理的get数 > 50k



服务器端缓存（3）

- 全局数据
 - 请求数量大、内容完全一致、变更频率低
 - Memcache
 - 缓存命中要求100%
- 一般局部数据
 - 请求数量较大、内容不一致，但很多组成部分一致、变更频率较高
 - Redis+轻量计算
- 其他数据（数据长尾部分 + 数据写入操作）
 - 请求数量不大、内容完全不一致，变更频率高
 - 对系统性能压力大
 - Redis + MySQL + 前端计算
 - 极端情况需要逐接口特殊对待
 - Ex: 按昵称查找用户



服务器端缓存（4）

- 全局数据的缓存永不过期
 - 主动Update取代过期Update
- 缓存调整的关键在于数据可视化
 - 逐接口缓存命中率
 - 命中层次
 - 命中后的性能
- 灵活根据业务需要设计缓存
 - 是否需要缓存？哪一层的缓存？缓存时间？
- 应急响应的核心之一在于缓存调整
 - 开发对应的接口调整各级缓存时间，用数据的准确性换系统吞吐率
- 跨数据中心缓存
 - 主数据中心 \longleftrightarrow 从数据中心



API statistic data on 59.151.4.128 on last 30 seconds

action	total count	API processed count	cache hits %	avg api processing time	avg non-a
uploaduserheadphoto	0	0	---	---	---
uploaduserphoto	5	5	0%	376.27ms	---
uploaduserwork2	3	3	0%	0ms	---
followuser	22	22	0%	79.7ms	---
cancelfollow	1	1	0%	4.26ms	---
registerclient2	37	37	0%	20.02ms	---
registerclient3	6	6	0%	13.34ms	---
register	11	11	0%	11.39ms	---
login	5	5	0%	2.13ms	---
gethottestsong	0	0	---	---	---
getartistlist	15	0	100%	---	3.29ms
getsonglist	54	0	100%	---	18.25ms
getuserphotolist	633	339	46.45%	5.88ms	1.71ms
getworkscorepercent	125	0	100%	---	1.45ms
getuserworklist	161	78	51.55%	45.42ms	1.74ms
getuserstatistics	148	148	0%	11.4ms	---
getuserevents	11	11	0%	121.08ms	---

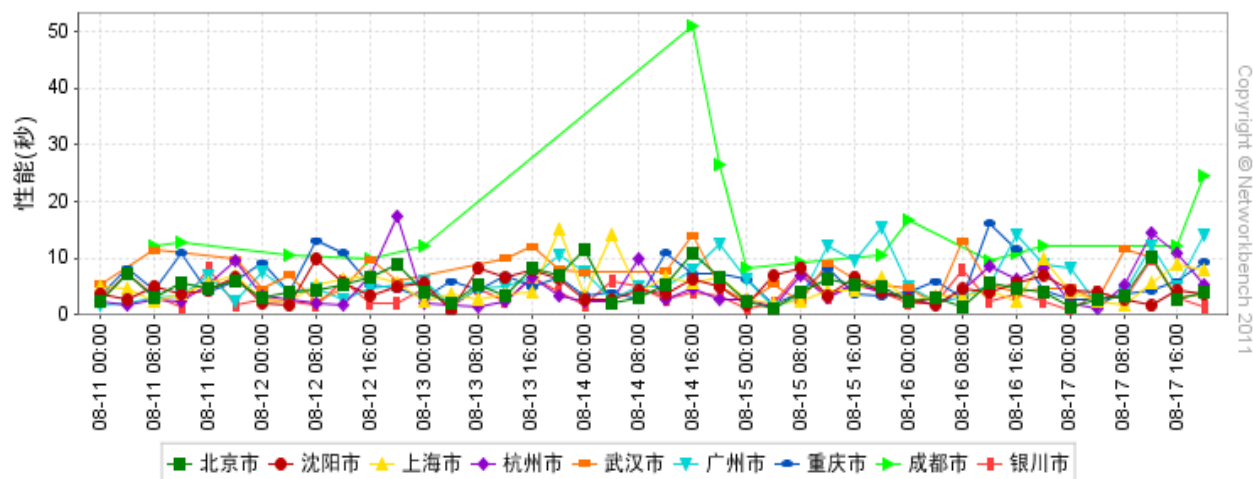


CDN的选择

- 为什么要上CDN？
 - 网络环境复杂
 - 移动、联通、电信、教育、GPRS、3G...
 - IDC价格比CDN高一个数量级

- 怎么选CDN？
 - 价格
 - 性能
 - 第三方评测

城市性能曲线图-性能(秒)(2012年08月11日 00:00 - 2012年08月18日 00:00)



CDN调优

- 节点分层
- 按照用户运营商比例调整节点数
- 第三方评测 vs 自统计
- 不要把鸡蛋放到一个篮子里

CDN的延伸

- 通用存储云 + CDN
 - 待分发内容直接上云
 - 解决回源压力
 - 解决数据备份问题
- 图片云 + CDN
 - 性能为图片专门优化
 - 图片防盗链
 - 自动转分辨率
- 视频云 + CDN
 - 性能为视频专门优化
 - 快速自动转码
 - 内容审核
 - 广告集成

Thanks!

