

# 微博cache设计谈

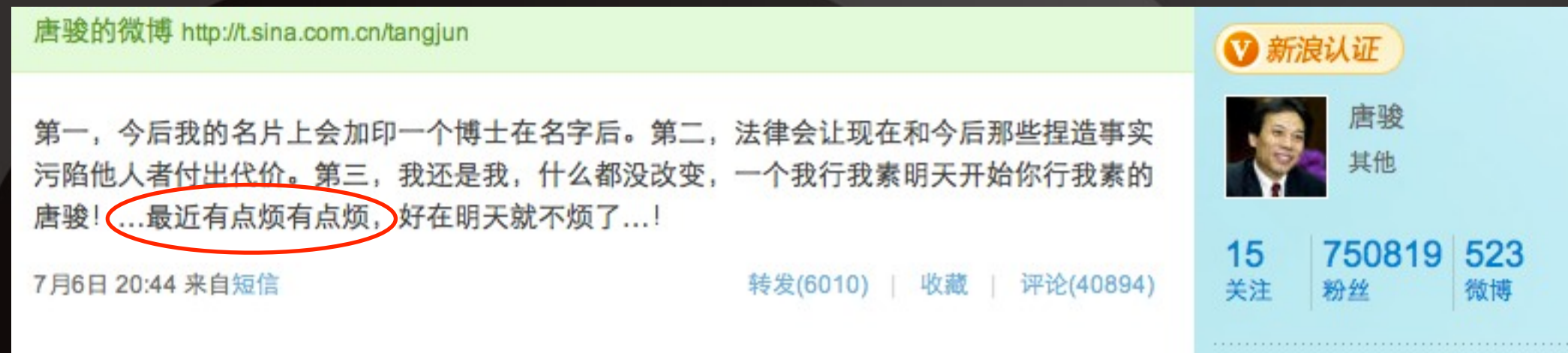
非最终版

# Agenda

- Feed架构
- 微博中常见Cache问题
- 扩展

# 微博技术的核心

- 数据的分发、聚合及展现
- 每条微博, 在技术上也称为status或feed
- 如



# Feed架构

- 微博两种feed设计模式
  - Push(推)
  - Pull(拉)
  - 其他

# Push

- 将feed比喻成邮件
  - Inbox: 收到的微博
  - Outbox: 已发表微博
- 发表: 存到所有粉丝inbox(重)
- 查看: 直接访问Inbox(轻)
- Offline computation

# Push

- 优点：实现简单，首选
- 缺点：分发量



The screenshot shows a Weibo profile for Yao Chen (姚晨). The profile includes a cartoon avatar of a yellow egg character, the name '姚晨' with a verified badge, and her location '北京, 朝阳区'. The number of followers is 2133375, which is circled in red. Other statistics shown are 510 followers and 1934 tweets. The profile also includes a bio, a 'Follow' button, and a 'Recommend to friends' button.

关注	粉丝	微博
510	2133375	1934

# Pull

- 发表：存到自己outbox(轻)
- 查看：取所有关注对象Inbox(重)
- Online computation



# Pull

- 优点：节约存储
- 缺点：计算量大，峰值问题



# 共同的难题

- 峰值挑战


微博小秘书的微博 <http://t.sina.com.cn/sinat>

南非#世界杯#终于迎来了高潮中的高潮！德国与英格兰奉献了一个精彩的半场，流畅的进攻、历史性的误判、永不妥协的精神，无论如何这都将成为历史的经典！同时，每秒世界杯微博峰值达到2500条，再创围脖新高！围脖球迷朋友们，让我们为精彩欢呼吧！加油，德国！加油，英格兰！

6月27日 23:02 来自新浪微博

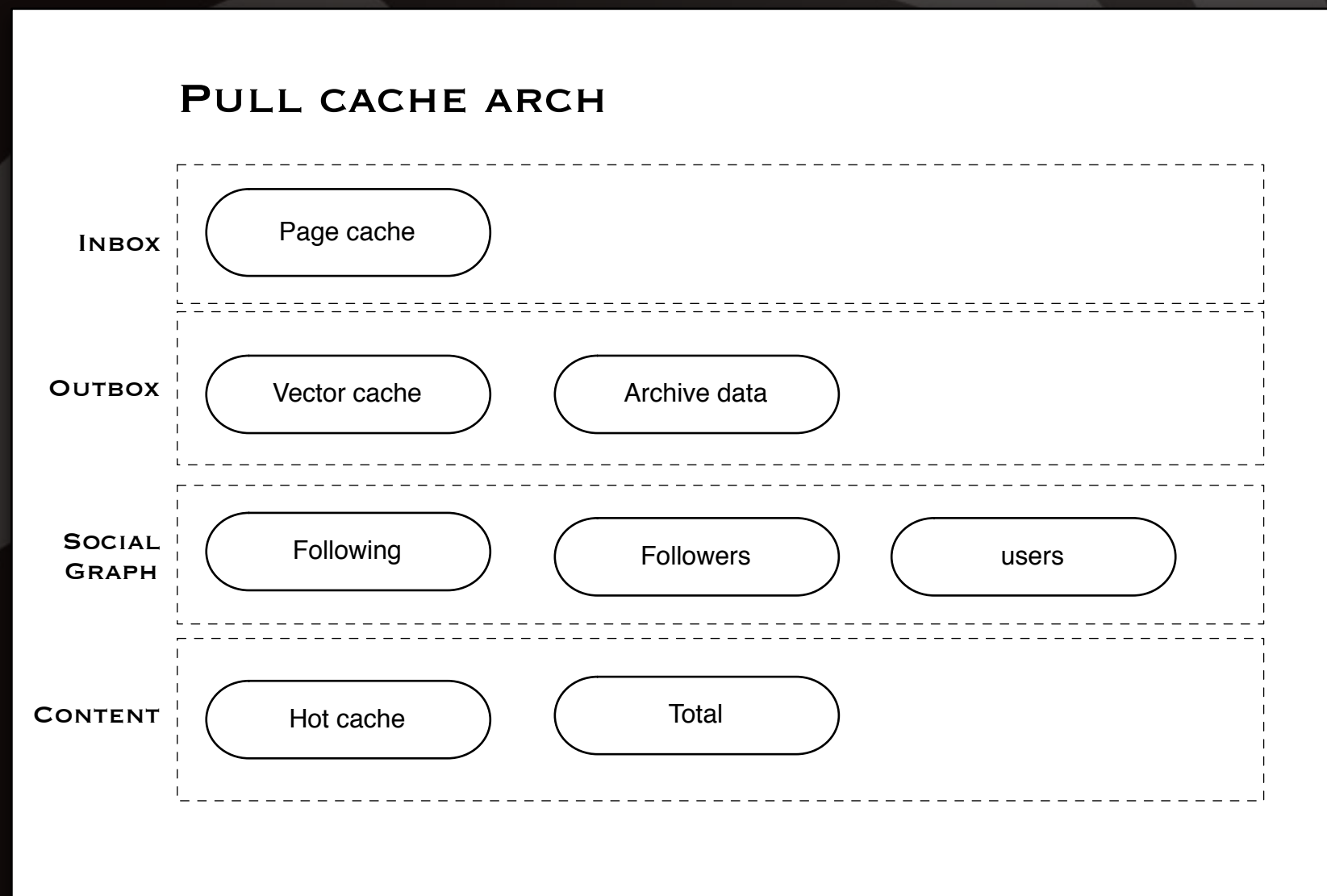
转发(248) | 收藏 | 评论(144)

**V 新浪认证**

 微博小秘书  
北京 海淀区

1979	4247588	1058
关注	粉丝	微博

# 典型的架构



# Cache

memory is the new disk,  
and disk is the new tape.  
for "real-time" web applications,  
and systems that require massive scalability

- Jim Gray

# Memcache 使用方式

# terminology

- content cache: 内容体
- vector cache: feed id list

# content cache

- multi get n ( $n = \text{items/page}$ , e.g. 50)
- $\text{size} = n * \text{length of feed} + \text{header}$  (e.g. 2k)
- concurrent requests / sec, e.g. 1,000
- $\text{total} = 50 * 2k * 1000 / \text{sec}$
- $\text{total} = 100M * 8\text{bit} = 800\text{Mbps}$

# 问题



# 1. 带宽

- 1,000并发，需要800Mbps带宽
- 1万并发，需要8Gbps
- facebook Memcache: 60GB ~ 120GB/s

# 1. 带宽

- 在IG内网，只能压力到 300~400Mbps
- 后来将热门数据加载到local cache
- 压缩
- 复制

## 2. 压缩

- gzip
- quicklz
- lzo
- 速度比较数据, zhangwei

# 3. 连接数

- 1台前端配置100连接
- 如果有100台前端，则memcache高达10万连接
- 考虑memcache udp协议
- Throughput/Latency/connections
- todo: 比较数据, zhangwei

# 4. 容量

- 将不同业务、不同长度的key存储到不同的memcache池
  - 不同的业务有不同的生命周期
  - LRU cache, 小量
  - Memory storage, 大部分
  - 更高效的内存利用

# 5. hot keys

- content cache of 姚晨
- create local cache
  1. get user\_yaochen\_local
  2. get user\_yaochen
    1. set user\_yaochen\_local:value
  3. 删除时需要delete all

# 6. mutex

- 姚晨的content item未cache
  - 每线程需要从db加载数十条微博数据
  - 上千个并发线程同时穿透到db
- key\_mutex 先添加成功，再更新key
- mutex 失败则等待，重试
- 添加结束，删除mutex



# 6. mutex

- 场景2: 姚晨的粉丝列表cache
- 更新太频繁，刚构建就被删除
-

# 7. value list

- vector cache, 假如放100条最新
- {1,2,3,4...98,99,100}
- 如何高效更新?
- append/prepend只能添加, 不能自动丢弃老的
- 需要应用需要get读出合并再set

# 8. memcache proxy

- 为什么需要一层proxy
- pros
- cons