

# 微信架构的启示

Wednesday, May 16th, 2012 by Tim | Tags: qq, weixin

腾讯大讲堂中最近分享了周颢演讲的微信技术总监解读微信架构的秘密，看完视频的一些心得。



## 技术微创新

微信的技术设计上有很多微创新，看起来都很小，但是对于系统的稳定性、用户体验及开发敏捷都具有重要作用。

### 前轻后重

由于客户端升级不便，从技术设计上尽量利用后端的设计来减少依赖客户端升级的方法。如某个版本新增了群聊功能，按常规思路，需要所有客户端升级才能全部打通。微信采用服务器兼容的方法，在老客户端不升级情况下让其增加群聊的功能，通过在服务端将群聊协议转换成之前旧版兼容的协议返回给老的客户端。

### 客户端辅助设计

微信客户端做了很多非常规的功能，比如常规的客户端测速方法是登录阶段轮询测试多个IP来选择服务器，这样会带来流量及登录速度双方面的开销，因此微信选择的方法是服务端返回最佳的IP（可能是通过历史数据分析）。客户端另外实现了一些容灾能力的配合，当一个IDC访问出现异常自动选择另外一个IDC。

### 流量控制

由于大部分无线用户对流量非常敏感，为了防止由于客户端不可预知的bug如死循环导致“偷流量”，服务端增加用户流量实时分析的方法，可以在海量数据下找出流量异常的用户，并给这些用户强制下行终止连接信号。

## 基础研发的策略

从视频介绍来看，微信的基础研发与业务结合非常紧密，提到的有基础组件比如 Client/Server 框架，从原理上看类似 Twitter [Finagle](#)，框架封装了大容量网络通讯及远程调用所需各种功能。此外介绍的一些监控与统计框架也较为先进，可以随时增加监控指标。

大量可复用的基础组件让业务开发非常敏捷，周颢总结的基础研发策略是“实现已有经验的固化”。这和其他一些公司中的基础研发团队思路有所区别。业界中基础研发脱离生产闭门造车的情况并不罕见，一方面业务部门重复低层次开发现象严重，另外一方面基础研发对业务理解欠缺，开发的组件模块与业务结合不紧密，无法被有效利用。因此类似微信这样增强业务团队的力量，在开发业务同时总结抽象更多的基础组件或许是一种更为实用的发展思路。

## 腾讯海量课程

视频中多次提到腾讯内部的一种海量模型的培训课程，其中的经验或设计模式包括 **大系统小做**

将一个复杂的大系统分解成多个独立的、小的、简单的任务去完成，“分而治之”。

### 柔性可用

服务端系统通常不是 0 与 1 之间的选择，可以在极端情况下做一定优雅降级，在服务端代码中需要体现这些设计。

### Set 模型

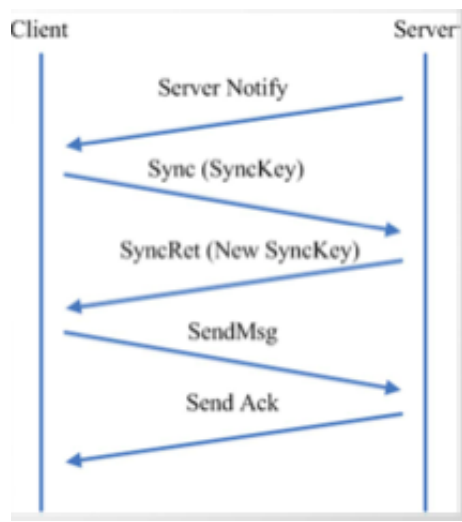
类似最近讨论的 [cell 架构](#)，按一个服务按用户范围分成不同的小单元，每个小单元 (cell/set) 具备全部业务服务能力，当一个 set 发生故障时候，只会影响这一小部分用户。微信架构中所做的补充是，在 set 之间增加了容错处理，当一个 set 发生故障时，使用类似一致性哈希的算法，调用方可以自动切换到下一个 set 来存储，并且将新的位置记录在 index 上 (类似 GFS master)，周颢自称是一个简化版的 GFS。

## 微信的协议

里面提到了 XMPP 和类 Sync 的自定义协议，里面提到 XMPP 的缺点是流量大和消息不可靠。但是流量大并非 XMPP 主要矛盾，可以很简单将其映射成二进制协议。消息 ACK 也可以添加简单的扩展协议来实现。较繁琐的还是兼容 CMWAP 网关的设计。

使用 XMPP 或者简化的 XMPP 标准协议有很多好处，类似的场景有业界广泛使用的 open api 基本都使用 HTTP 及 JSON，并不是由于这两种协议优化或高效，而是其简洁并得到广泛的认知。一种标准协议的认知及扩展成本要比一个自定义协议小得多，

XMPP流量大的问题可以通过转换协议来实现，比如用binary 1代表login等全部协商协议，2代表message，消息增量获取也可以通过自定义扩展协议来实现。标准协议可以让团队内部及新人的认知成本降低，每一个参与者都很容易想到代码及架构改进建议。而且微信目前也在构建开放平台，自定义协议在开放方面必然具备一些局限。



## 其他

### 分布式理论

从微信的分享也看到，指导业界不同背景的团队（不管大小）的分布式理论主要还是Google及Amazon系列论文，对互联网技术的发展具有深远影响。微信借鉴了Quorum及Merkle tree的理论，创造了一种自定义的做法，用于实现一种分布式递增发生器。

不过分布式递增算法其实有更简单的实现方法，twitter也有相关的公开博文，由于视频相关背景介绍比较简短，可能大家解决的需求具有差异，就不展开了。

### 监控

数千的监控项，可以在分钟发现系统的异常

### 敏捷

每天20个后台变更

### 技术传承

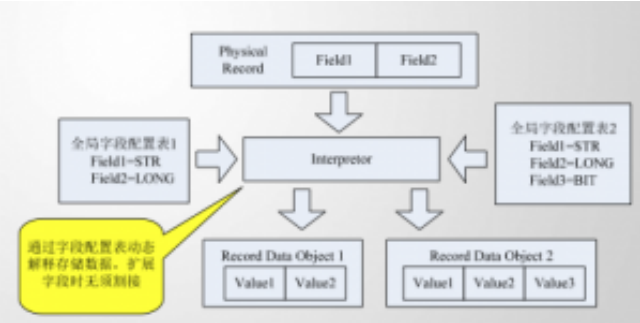
从视频和PPT来看，微信的技术体系是从头搭建的，可能有不少利用qqmail时代的基础代码，但与深圳团队并不存在太大技术沿袭或者传承关系。从另外一个角度也看到微信技术团队的战斗力。

### 新人力量

一位毕业生的创意：按SET分布,全量数据从2G减到200K（具体的情况待了解）

另外一个新特性，漂流瓶及摇一摇，据说也是2个月的本科毕业生一周完成，而且上线后运行稳定。

存储




上图中可扩展设计中字段配置表的做法感觉略显繁琐，目前大部分NoSQL产品的schema free方式可能更易于维护。

产品驱动

如图，其中“允许发布十分钟前的变更”这样做法通常在大型团队通常会引起广泛质疑，单纯学习这种形式并非正解，如何在互联网产品上实现敏捷值得产品和研发进一步思考。

项目的敏捷



- 敏捷就是试错法
- 敏捷是一种态度
  - 产品决策是成功的第一因素
  - 允许发布前十分钟的变更
  - 给予产品决策以最大自由度

腾讯大讲堂 <http://djt.qq.com>