

千万级在线推送系统 架构解析

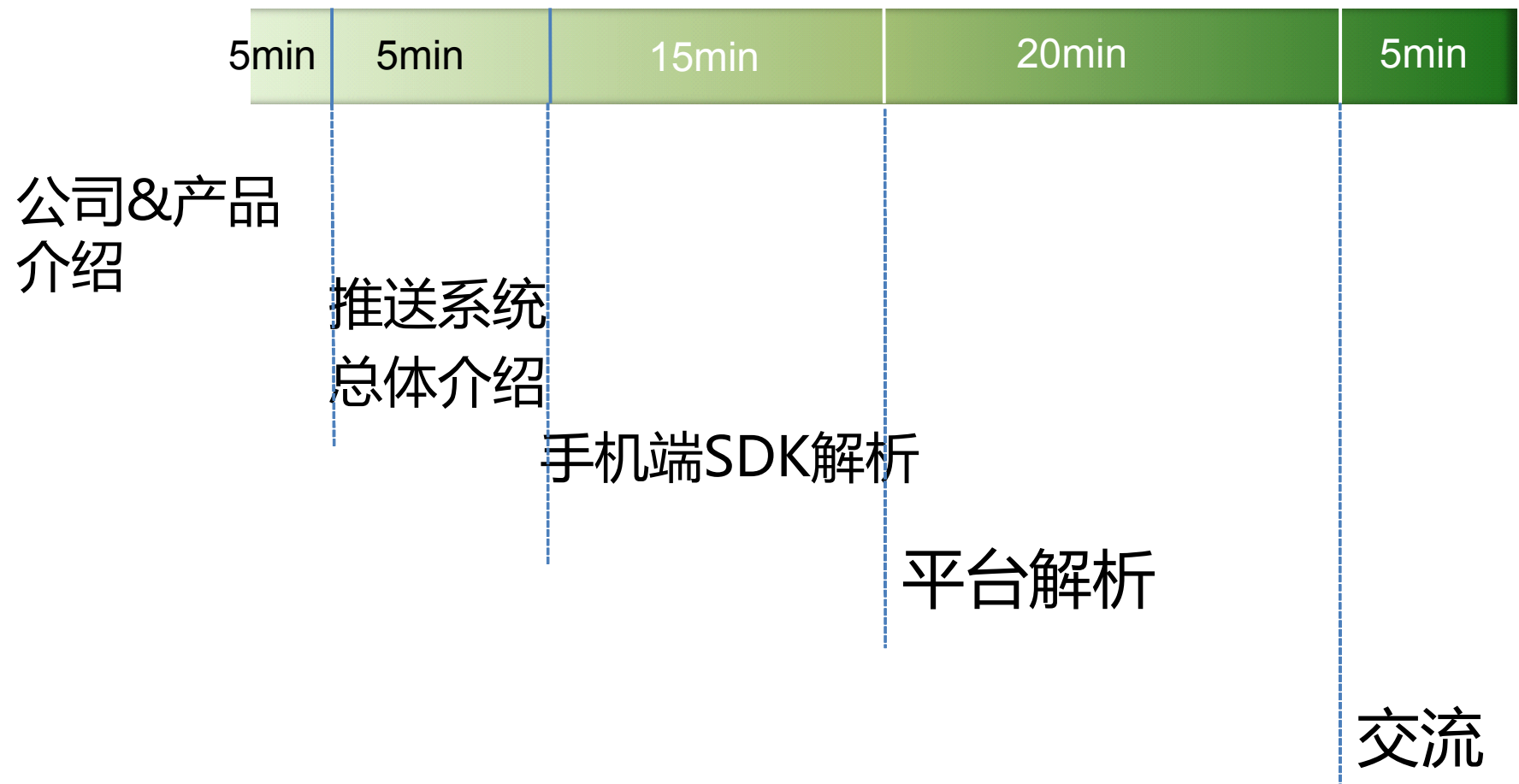


叶新江
Anson

@猪立叶-Anson
@个信互动
@八爪推送

Agenda

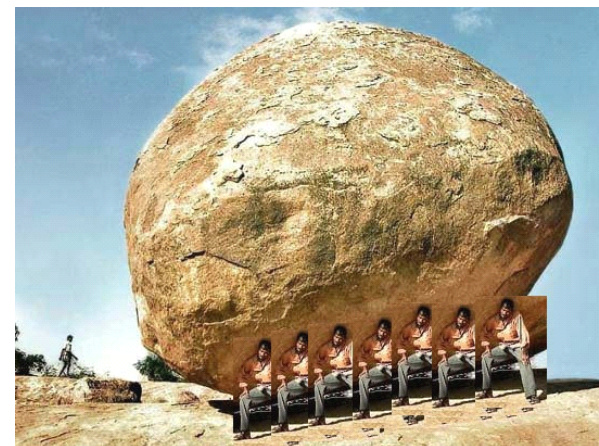
Total Est. 50min.



公司及产品简介

Found @ **2010年9月**

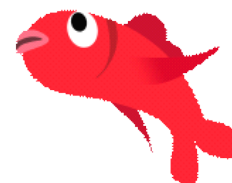
一起扛着这个公司的人 **~100** 人



个信



八爪



个推



个信

有事没事来“个信”儿!



个信 有事没事来”个信”儿!

2011 年 1月发布首个版本，是同类产品中发布最早的

主要功能：

-好友（通讯录中）之间**免费**发送短信和彩信

-**WEB端**/PC客户端收发短信和彩信

-**智能拨号**，英文/中文/拼音一搜到底

-通讯录、短信内容**备份/恢复**

-第三方业务短信彩信和应用的**发布通道**

-支持Symbian, Kjava , Android , MTK , iPhone , Window Mobile等所有主流手机平台

<http://www.igexin.com>



iBazhwaTM

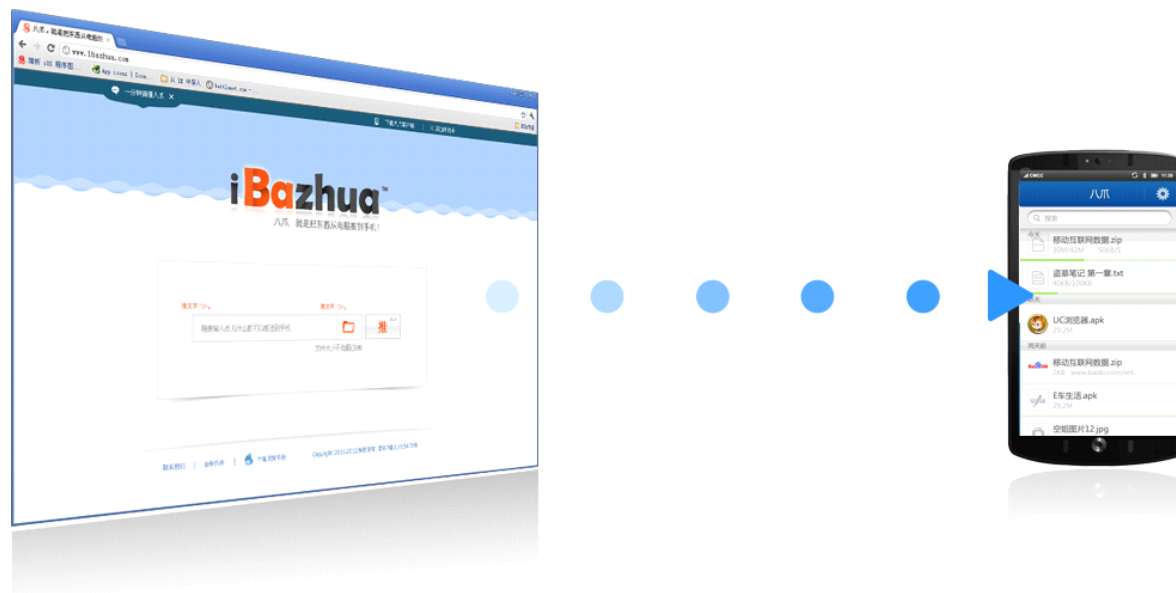
八爪，就是把东西从电脑推到手机！

个人PC与移动设备



- 数据无线推送
- 随时并高速地将PC内容 (文档/视频/音乐/照片/应用...) 推送到手机

互联网与移动设备

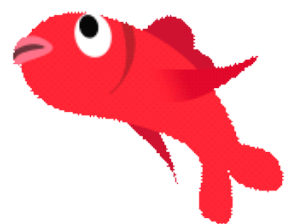


- 任意网站信息与手机的推送
- 通过八爪API或浏览器插件，一键推送互联网内容到手机
- 网站植入八爪JS代码，推送到移动设备

与第三方互联推送



- API开放
- 八爪™ 认证账号与推送能力开放



推送系统

以上产品均有推送系统的基因在里面。

接下去我们进入正题 -----

移动互联时代的推送诉求

SMS 推送时代被证明了推送的价值

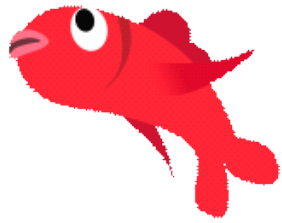
但是：

- 内容不够丰富
- 大小限制
- 无法满足端对端的完成业务链

移动互联时代的推送诉求

我们需要的是：

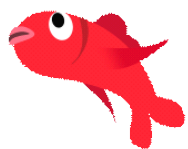
- 更低成本更高效率, 最好不要钱
- 展示的内容丰富，有声有色最好还有视频
- 能够智能判断内容对应的应用，没有应用的话，能自动下载和安装
- 和应用能互动, 能激活应用
- 对用户分段，针对性要强
- **(总之要求是一个神器)**



个信推送平台（个推）

a push service, not only notification but also action and engagement

- 目前针对 Android 和 Symbian, 当然主要是 Android 了
- 前端 4S (slim, save power, save traffic, stable)
- 创建和应用的互动和带动 (action & engagement)
- 千万级并发支持



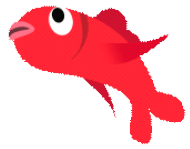
个推 – 手机端（1）

- **SDK 的 4S (slim, save power, save traffic, stable)**

这个很重要，因为你不能喧宾夺主。

个推 SDK 的几个数据：

- 空载连接每月流量消耗 < 2M
- 电量消耗 < 2%
- SDK 安装包空间占用 < 200KB
- 正式发布到目前为止， 无 crash 报告



个推 – 手机端（2）

- **4S (slim, save power, save traffic, stable)**

不要用写 PC 程序的方式来写 Mobile 应用。

需要及时处理消息，合理避免手机CPU sleep.

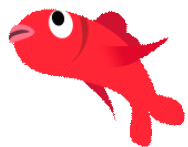
(wake lock / alarm manager)

采用尽量少的线程，用类似 ActiveObject 的方式来用单线程模拟多任务实现

合理的 heartbeat - 4 min.

使用私有二进制协议代替 XMPP，消息Bundle发送

独立的进程，尽量不影响应用本身



个推 - 手机端 (3)

- 内建的 Content Render Engine (CRE)

布局

内容抓取

Text, Image, Web View, Video

- 内建的 Action Chain Engine (ACE)

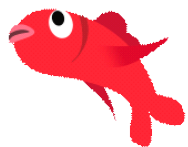
目前使用 JSON 方式定义 Action Chain

正在实现 VM 方式

Action 类型(列出其中一部分) :

- 消息透传
- 调转到 browser
- 启动应用
- 下载应用并启动安装
- 激活应用





个推 - 服务端 (1)

- 推送方式的选择

Pull - 轮询

频率的选择是关键

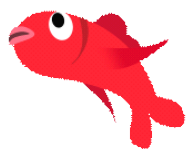
Pull - BOSH/Comet

基于 http 实现, 较复杂

大并发下的效率是个问题

个推采用 Push 方式。

- * 实时要求高
- * 数据不常变化
- * 在解决省电省流量基础上



个推 - 服务端 (2)

- 并发接入能力

尽可能大的单台设备的接入能力是大用户量系统必须解决的问题。

据传：

2011/09, WhatsApp 100 万连接

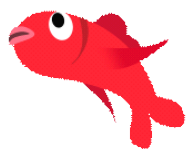
2012/01/06, 200 万TCP连接

Urban Airship 在2010年 50万连接

个信：

单个 Java JVM 4G 实例，支持 50 万个业务连接（非简单连接）

2012 年目标，单台支持 100 万并发连接



个推 - 服务端 (3)

- 大并发长连接下需要考虑的关键点

尽量减少应用内存的footprint.

高效使用内存(pre-alloc mem pool, thread-local-mem pool)

选择合适 OS 和 语言组合(FreeBSD, Centos with Erlang, Java, C++ etc.)

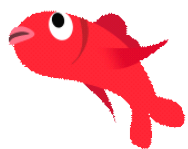
如果是 Java, NIO 是必须的.

NIO 事件处理模式的选择 (Same Thread, Worker, Dynamic, Leader-Follower).

如果是 Java, JVM 的调优也是关键 (Heap size, GC etc)

注意一些已知的问题 (譬如 JDK Select Spin)

内核调优 (特别是网络参数方面)



个推 – 服务端（4）

- **异步系统**

要增加系统的吞吐量，系统内部需要异步调用
复杂度增加

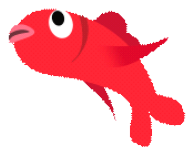
最好需要内部有统一的通讯框架(支持同步、异步、回调；广播、随机、轮询)

- **分布式缓存**

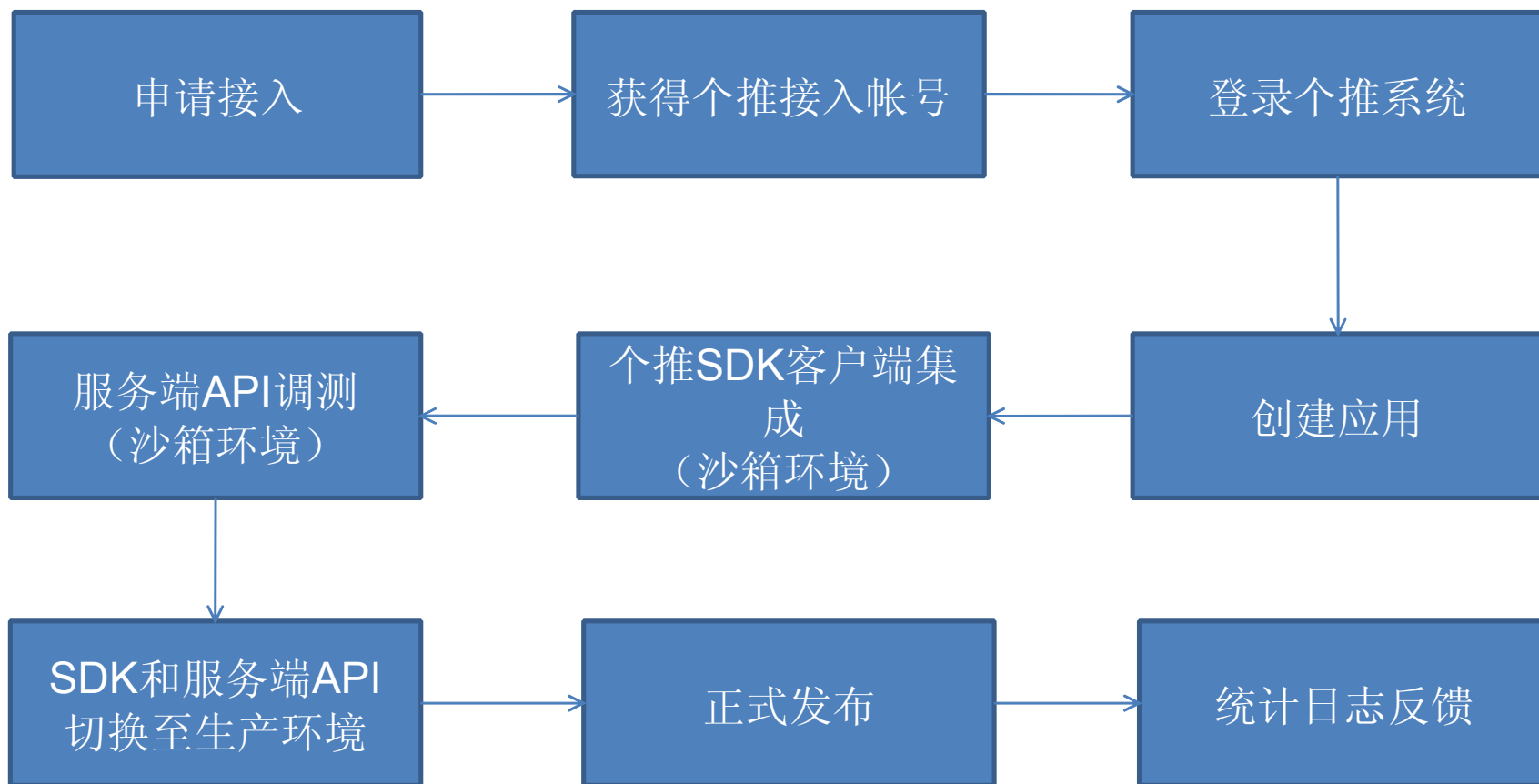
自动平衡、自动复制、容错（可以借用目前使用的一些分布式缓存的 cluster 方案）

- **数据分析和 BI**

大数据量处理（Hadoop, Hive, infoBright w/ MySQL）



个推 - 如何接入 (@钱昊_个信)



Thanks , End!



Contacts:

mailto: exinjye@gmail.com

weibo: @猪立叶-Anson

msn: yexj_zj@hotmail.com