

[iOS开发](#) [Swift](#) [App Store研究](#) [产品设计](#) [应用评测](#) [游戏开发](#) [苹果相关](#) [WebApp](#) [安卓相关](#) [营销推广](#) [业界动态](#) [程序人生](#)

发现一种创新的开发方式

更智慧的app开发来自IBM

免费试用 IBM Bluemix™ 提升你的app开发效率 →

发现一种创新的开发方式  
更智慧的app开发来自IBM

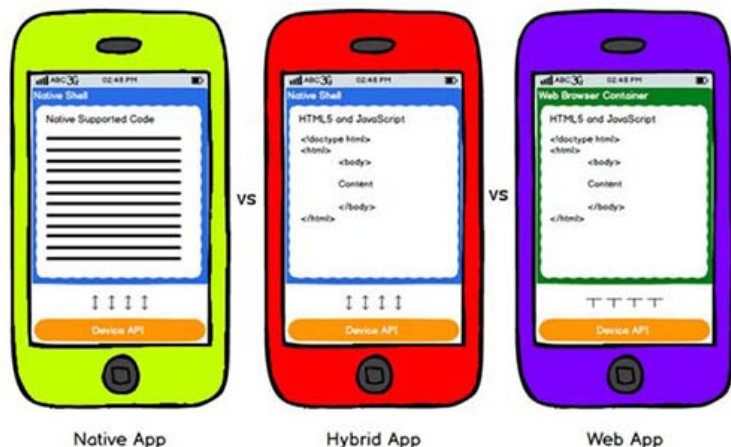
免费试用 IBM Bluemix™ 提升你的app开发效率 →

[首页](#) > [iOS开发](#)

## iOS hybrid App 的实现原理及性能监测

2015-11-18 11:39 编辑: lansekuangtu 分类: iOS开发 来源: CocoaChina

1 1107

[iOS](#) [原理](#) [hybrid app](#) [性能监测](#)[招聘信息: 手游JAVA 服务器端主程](#)

作者董一凡自述: 作为一名写了十年代码的程序员, 目前我最擅长的领域是移动平台的客户端开发, 在移动领域的开发时间超过七年, 前前后后涉猎过很多个平台。随着大部分移动平台自己走向死亡, 现在我也主要专注在了iOS和Android两大移动平台, 偶尔也会客串下Windows这个不知道是移动还是桌面的平台。十年前, 我刚入行的时候, 曾经认为自己将会永远做一个C++程序员, 于是花了大量时间在C++上。现在C++也是我工作所用的主力语言之一, 工作之外也会偶尔写点什么娱乐一下。写了一些年程序后, 终于意识到了之前定位的狭隘, 于是开始广泛的学习各种技术, 各种各样的语言也学了很多, 值得庆幸的是, 几年折腾下来, 我一直也没有对写代码这件事感到厌倦, 于是我又认为自己将会永远把开发做下去。现在, 我也觉得开发是一个可以终身做下去的事业, 不过除了事业我还想追求更多的东西, 从这些年的经历来看, 其中贯穿始终的就是在不停的学习, 想明白这一点后, 我开始除技术之外更广领域的学习, 比如日语, 画画, 设计, 钢琴等等, 给自己的定位也变成了在今后作为一名终身学习者。

### — iOS hybrid App 简单介绍

大家应该多少都知道, iOS 设备上有两种入口, 一是通过 App Store 下载一个个的 App, 另一个是用系统浏览器去访问网页。前者我们一般称为原生应用, 后者就是传统意义上的网页。两者各有特点, 开发一个原生应用, 一般是使用 Apple 给我们提供的开发工具和 Cocoa 框架。优势就是可以利用到系统的所有特性, 做出很酷的特性而不损失任何的性能, 而缺点就是每次 App 提供新功能都必须重新打包 App, 提交给 Apple 进行审核, 通过以后再上架 App Store, 最后用户再升级, 平均需要两周的时间。相反, 写一个网页则完全没有这个限制, 服务器做一次升级, 用户通过浏览器再访问, 就是最新的了, 而写网页的缺点则是受到很大的限制, 很多系统特性是无法访问的, 而且性能往往

### 热门资讯



这些 iOS 面试基础题目, 你都深入了解吗?

点击量 16507



【译】17个提升iOS开发效率的必用工具

点击量 10629



iOS开发——UI组件 (个人整理)

点击量 8431



我已经写了48年代码了, 我感觉我还能写下

点击量 5436



出大事了! A商要联合程序员们打淘宝分京东

点击量 5303



ViewController 瘦身 的另一种解决方案

点击量 5070



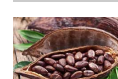
新手向: 五分钟搭建 App设置页面 纯代码

点击量 4933



来就来全套! 敏捷开发知识体系笔记

点击量 4443



让我们来搞崩 Cocoa 吧 (黑暗代码)

点击量 4388



XcodeGhost S: 变种带来的又一波影响

点击量 4104

### 综合评论

“ 在使用UITableView和UITableViewCell的时候一定要考虑到cell被复用的情 easingboy 评论了 Objective-C 编码建议 ”

感觉unity满好玩的, 跟cocos2d不同感

ilovemotw 评论了 Unity2D 开发 + UGUI 系列教程 (二) ...

看上去似乎虽然对用户跳失率没影响, 但是却有一个更严重的问题: 卸载。不花飞花满天 评论了 无线端两种弹窗广

不高，以至于很难实现一些很酷的效果。

鉴于原生应用和网页各有优势，所以就衍生出了一种介于两者之间的开发方式--混合应用（hybrid App）。其特点是在原生应用中嵌入一个浏览器组件，然后通过某种方式，让原生代码和网页能够双向通讯，结果就是可以在需要原生功能的时候使用原生功能，而适合放在网页端的部分就放在服务器上。某种程度上利用到了两者的优势。另一个优势就是，由于网页技术在 iOS 和 Android 上是一样的，所以网页的这部分也就天然可以跨平台了。

二 如何实现 hybrid App

实现一个 hybrid App 最简单的方法就是使用 [Apache Cordova 开源框架](#)。Cordova 已经帮你做好了所有的网页和原生应用之间的桥接工作，你需要做的就是根据他的文档去写对应的网页代码和原生代码就行了。具体请参考[官方网站](#)

可惜的是，我们总有些场景无法使用 Cordova，比如我曾经的一个项目，项目主要是要提供一个 SDK，SDK 本身要使用 hybrid 的技术。但是 SDK 的用户可能也会用到 Cordova，有些情况下，两者用的 Cordova 为不同版本，正好无法兼容。于是就需要自己去实现 hybrid App 的底层了。

三 iOS hybrid App 的底层实现

1. 原生代码调用网页中的 JavaScript 函数

假设我们的网页中有如下代码

```
[script type="text/javascript"]  
  
    function myFunc() {  
  
        return "Text from web"  
  
    }  
  
[/script]
```

原生代码可以用如下方式调用 myFunc()

```
NSString * result = [self.webView stringByEvaluatingJavaScriptFromString:@"myFunc()"];
```

在这里 result 就等于 Text from web

2. 网页中的 JavaScript 调用系统的原生代码

这一步比上边的要复杂一些，iOS 不像 Android 可以直接给网页中的 JavaScript 函数注入一个原生代码的接口。这里我们会用一个比较曲折的方式来实现。

假设 Objective-C 的类里有一个方法

```
-(void)nativeFunction:(NSString*)args {  
  
}  

```

JavaScript 里我们用下边的方法来最终调用到上边这个方法

```
window.JSBridge.callFunction("callNativeFunction", "some data");
```

在我们的页面里，是没有 JSBridge.callFunction 存在的，这一步我们要在原生代码端注入。

在 webView 的 delegate 的 -(void)webViewDidFinishLoad:(UIWebView \*)webView 里我们用下边的方式注入 JavaScript

告方案的A/B test...

滴滴

dengwei 评论了 无线端两种弹窗广告方案的A/B test...

有没有 滴滴 弹窗的动画 大神求助

xu515835632 评论了 无线端两种弹窗广告方案的A/B test...

学习下，很不错啊，赞个

小小而已 评论了 Objective-C 编码建议

任意一行代码不超过80字符，这个有很多函数名字就会很长啊  
王尼玛 评论了 Objective-C 编码建议

回帖是一种美德，分享是一种精神！  
zhjl 评论了 轻松学习之 最简化原生效果集成UITableView...

iOS8真的是开放了好多接口，以后直接面向ios8开发能省事不少！  
king2142 评论了 轻松学习之 最简化原生效果集成UITableView...

感觉一个很基本的东西被你用名著“引用”装了一下b  
buffery 评论了 谈谈组件封装的思路和实现--PSCarouselV...

相关帖子

建议cocosjs团队解散，ios微信崩溃，qq浏览器崩溃，内存莫名其妙暴涨

那个泰然网发布的quick3.6社区版的帖子哪去了

照片相册管理器，多图上传，添加，删除，移动代码

调用相册失败，求大神拯救！！

怎么将下载到沙盒的视频保存到相册？

发现一个神奇的现象,swift

navigationController push 过去之后，pop回来时一个空白页面

有没有谁做过ios上的图书管理系统啊

UIWebView视频播放问题

```
-(BOOL) processURL:(NSString *) url
{
    NSString *urlStr = [NSString stringWithString:url];
    NSString *protocolPrefix = @"bridge-js://invoke?";
    if ([[urlStr lowercaseString] hasPrefix:protocolPrefix])
    {
        urlStr = [urlStr substringFromIndex:protocolPrefix.length];
        urlStr = [urlStr stringByReplacingPercentEscapesUsingEncoding:NSUTF8StringEncoding];
        NSError *jsonError;
        NSDictionary *callInfo = [NSJSONSerialization
                                   JSONObjectWithData:[urlStr dataUsingEncoding:NSUTF8StringEncoding]
                                   options:kNilOptions
```



UPYUN  
架构与运维大会  
Rediscover Arch&Ops  
2015.11.28  
北京

```
        error:&jsonError];

    NSString *functionName = [callInfo objectForKey:@"functionname"];

    NSString * args = [callInfo objectForKey:@"args"];

    if ([functionName isEqualToString:@"callNativeFunction"]) {

        [self nativeFunction:args];

    }

    return NO;

}

return YES;

}
```

从 bridge-js://invoke? 这个自定义的 URL 里边把附带在后边 JSON 字符串解析出来, 然后判断 functionname key 的值如果是 callNativeFunction 那么就调用原生方法 nativeFunction, 如果实现更多的方法调用, 只要添加这个映射关系就行了。

至此, JavaScript 和 Objective-C 代码的双向调用就都实现了。

#### 四 性能监测

Hybrid 和原生应用之间的争论一直以来都不少, 其核心问题其实是如何平衡开发成本和用户体验之间的关系。Hybrid 的开发成本一般来说要低于原生应用, 然后其体验总是要差一些。为了让 Hybrid 的用户体验能更可能的接近原生应用, 性能监测就显的更为重要了。

影响 App 使用体验一般来讲有两个主要方面

第一方面是 UI 的响应速度, UI 的流畅与否给用户的体验是非常不一样的。对这方面的性能监测, 一般的做法就是在主要的交互函数里打上时间戳, 而对于系统的 View, 也可以采用 Method Swizzle 的方法对所有的系统函数的调用时间进行统计。

二是网络, 而由于现在的大部分 App 都多少有了网络请求, 所以网络的请求速度也会很大程度上影响用户体验。网络问题在 Hybrid App 就体现的更明显。Hybrid App 总是会去加载服务器端的页面, 在页面加载出来之前, 很可能整个手机屏幕是空白的, 如果空白时间太长, 将是一个很糟糕的事情, 所以实时的监测请求网页的时间, 以及页面的加载速度就非常有必要了。针对 webView, 建议在它的 delegate 的几个方法里打上时间戳, 以此来统计页面请求和加载的时间。

总之实现起来, 并不是一个非常复杂的工作。然而性能监测的工作, 实现只是其中的一个方面, 由于用户的使用习惯, 实际的网络环境各种问题, 性能监测并不是在开发阶段监测一下就算完了的, 一般来说, 总是得把监测工作部署到最终用户的手机上去的, 如果是一个用户量不小的 App, 那么如何把收集到的大量数据很好的统计显示出来, 这完全是另一回事了, 把这事做好, 要牵扯到很多的数据组织, 前端展示的工作, 实际的实施绝对不是个简单的工作。

所幸, 现在已经有很多公司帮我们完成了这个工作, 比如 New Relic, App dynamics, Compuware, 听云等。这次我们就以听云为例, 看看他们是怎么来做性能监测这件事的。

#### 五 听云探针 (iOS App版) 的使用

听云对 App 的性能监测使用起来还是比较简单的, 简单步骤如下

申请完听云的账户后, 在添加 App 的地方填写相关信息

1 填写基本信息

2 嵌码

✓ 完成

App名称:

App名称可以随时修改, 请使用字母数字和中文, 最长32个字符, 请不要使用重复的App名称。

App平台:

☐ iOS ☐ Android

App类型:

请选择

保存并继续

之后就会得到一个唯一的 App Key

1 填写基本信息

2 嵌码

✓ 完成

App Key: 

复制

然后下载听云的 iOS SDK 的 Framework, 拷贝到项目中, 注意添加以下 4 个额外的系统库

- CoreTelephony.framework
- Security.framework
- SystemConfiguration.framework
- libz.dylib

然后在 App 的 pch 文件中包含听云 App 探针的头文件

```
#import
```

最后将 main.m 中加入

```
[NBSAppAgent startWithAppID:App_Key];
```

代码一般为下边的样子

```
int main(int argc, char * argv[]) {  
    @autoreleasepool {  
        [NBSAppAgent startWithAppID:App_Key];  
        return UIApplicationMain(argc, argv, nil, NSStringFromClass([AppDelegate class]));  
    }  
}
```

```
}
```

这样整个集成工作就完成了。启动 App，如果在 Log 日志中有如下内容显示就表示代码集成成功

```
NBSAppAgent 2.2.2.1
```

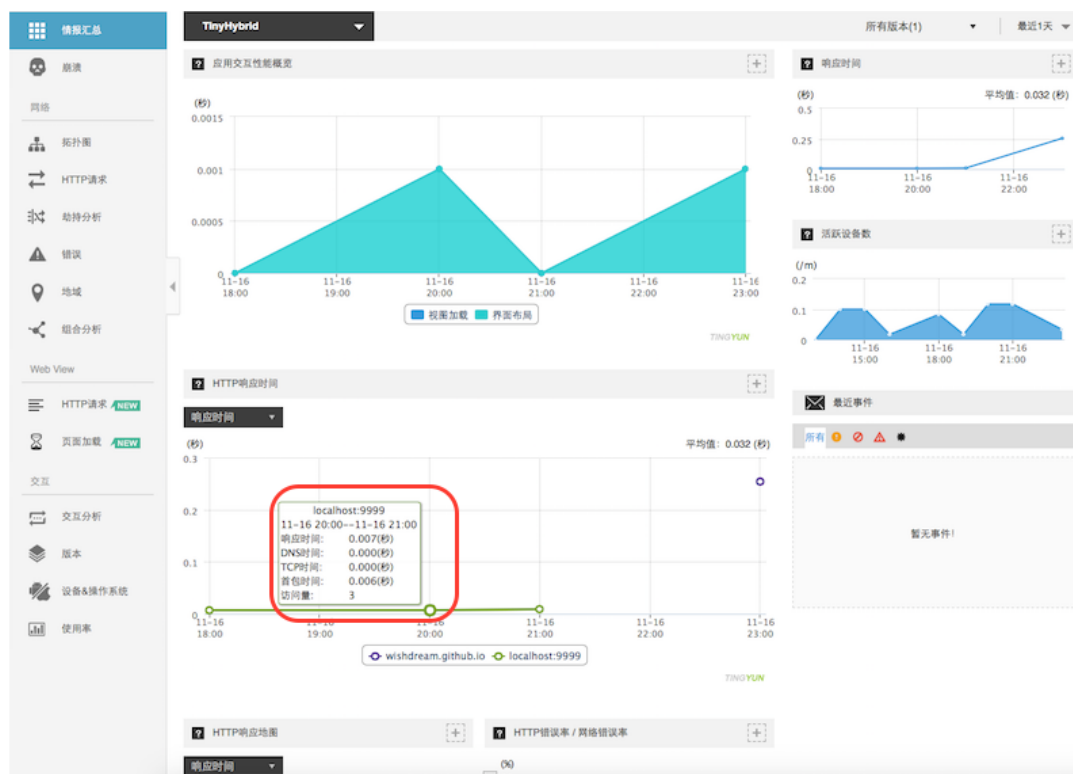
```
---->start!
```

```
Success to connect to NBSSERVER
```

## 六 听云监测数据观察

### 1. 汇总数据

登录到听云的后台管理页面，首先我们可以看到汇总的监测数据，图表的效果还是不错的，鼠标放到每个数据点上会显示详细的数据。



总体看来，分为两大类，一类是应用交互性能，这类主要是监测 UI 响应情况，会给出 view 加载，以及 layout 的时间汇总。如果发现某一项参数出现异常，那也许就是需要重构 UI 的信号了。另一类是网络性能，包含网络请求的响应时间等。

### 2. Web View

重要的东西最先讲，这个部分是听云目前最有特色的部分（好像是首家这么做的，目前还没在其他的类似服务里看到这个功能）。通常我们进行网络性能监测的时候，给出的是整个网络请求的情况，这在浏览器里边来说，整个网络请求其实也就是页面的请求，两者没有区别。而到了 App 里，同样是 http 请求，有可能是来自 web service 的调用，也可能是来自 web view 加载页面。而后者正好是我们讲的 Hybrid App 的主要实现方式。听云的这个条目就是完全只给出 web view 所进行的请求情况，换句话说，这是我们来监测 Hybrid App 网络性能的最好数据。

#### (1) HTTP请求



性能汇总

崩溃

网络

拓补图

HTTP请求

支持分析

错误

地域

综合分析

Web View

HTTP请求

页面加载

交互

交互分析

版本

设备&操作系统

使用率

TinyHybrid

所有版本(1)

排序 响应时间 隐藏<1%的吞吐量

最近6小时

性能一览

全部

自身服务

第三方

» wshdream.github.io 0.254秒

来源: 自身服务

» localhost:9999 0.008秒

来源: 自身服务

TOP5 响应时间最慢URL

(秒)

平均值: 0.035 (秒)

/\*.html (https)

/\* .html

0

0.1

0.2

0.3

20:00

20:30

21:00

21:30

22:00

22:30

23:00

TinyYUN

TOP5 吞吐量最高URL

(rpm)

/\* .html

/\* .html (https)

0

0.05

0.1

0.15

0.2

0.25

20:00

20:30

21:00

21:30

22:00

22:30

23:00

TinyYUN

这里所有 web view 所加载的页面的汇总数据。

(2) 页面加载

排序 页面加载 隐藏<1%的吞吐量

性能一览

全部

自身服务

第三方

TOP5 页面加载时间最慢URL

暂无数据!

听云除了给出网络性能的数据，这里还很贴心的给出了页面加载的汇总数据，要知道现在的网页是有可能非常复杂，包含很多页面元素的，在桌面端问题也许不明显，但在移动端，太复杂的效果也许会大大的拖慢加载速度，影响用户体验。根据这里给出的页面加载数据，就可以有针对性的去优化网页了。

3. 网络

这个条目主要是 App 的所有网络请求的数据。

(1) 拓补图

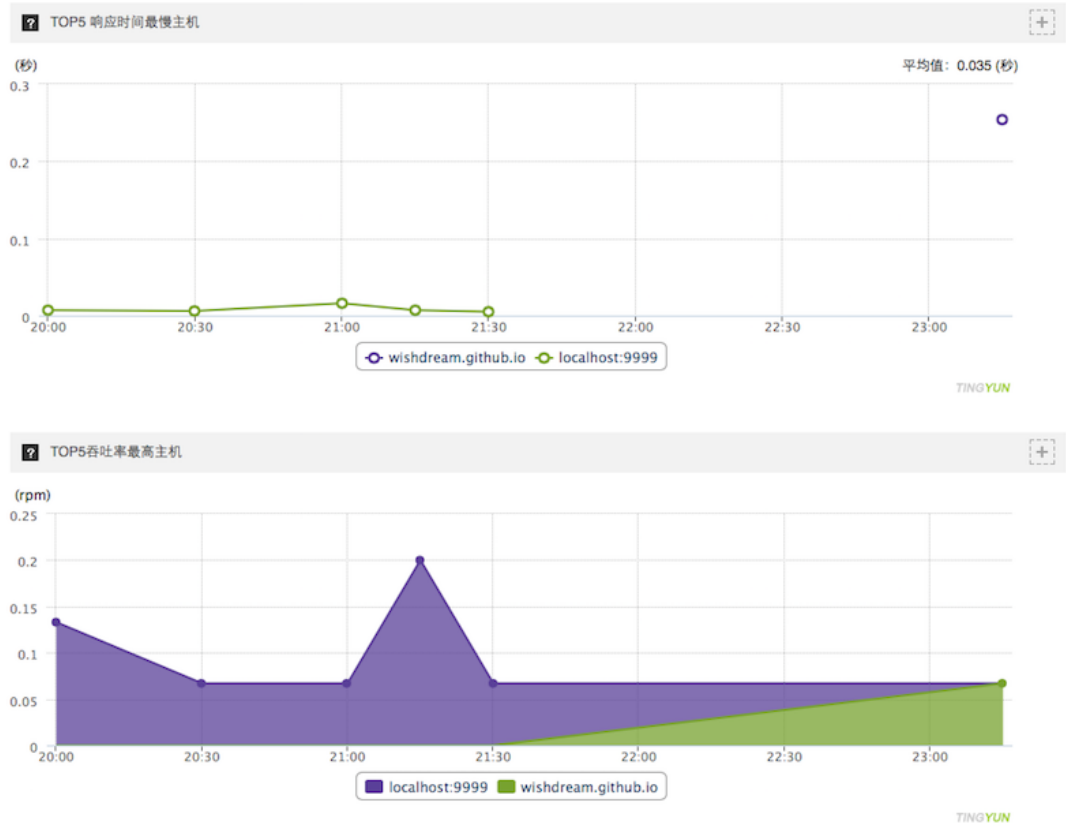
http://www.cocoachina.com/ios/20151118/14270.html

第 7 页 (共 12 页)



这主要是一个分类汇总的数据，可以分别查看是自己的 App 所以及第三方服务所发送的网络请求的汇总数据。

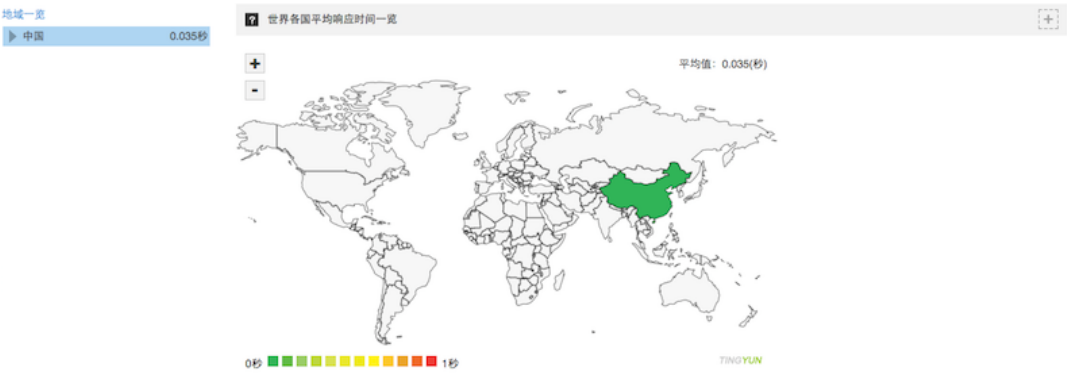
(2) HTTP请求



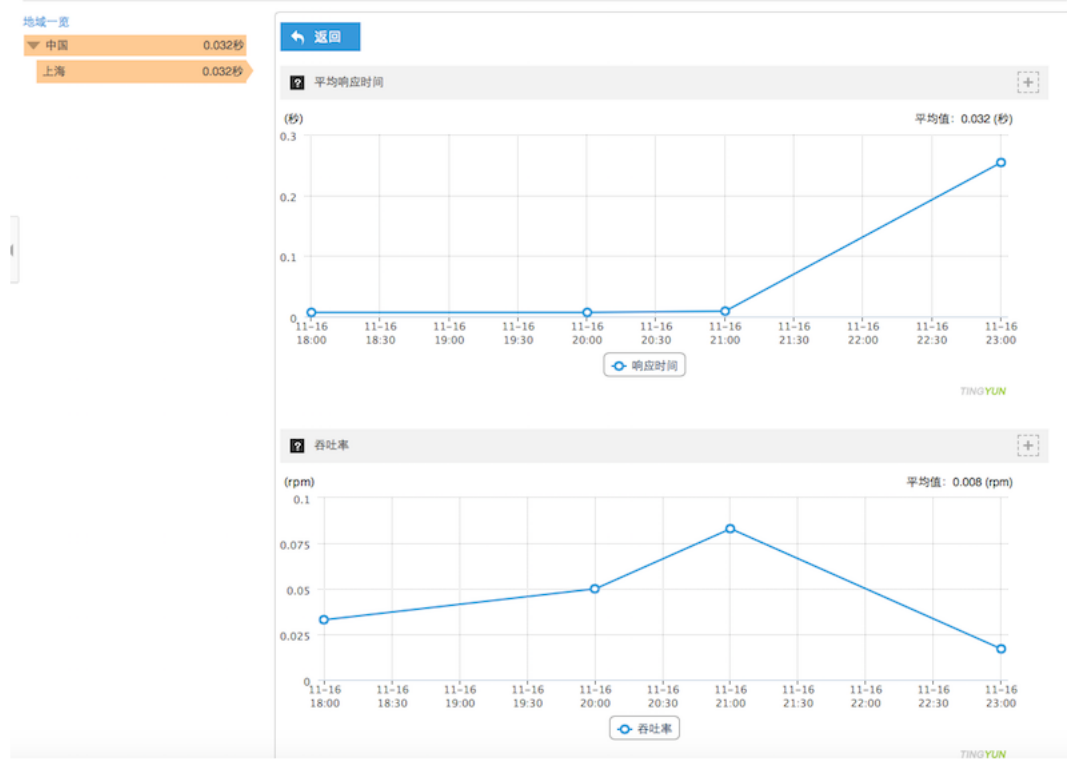


顾名思义，这里是所有 http 请求的详细数据，分别显示了响应最慢的主机，以及吞吐量最高的主机。

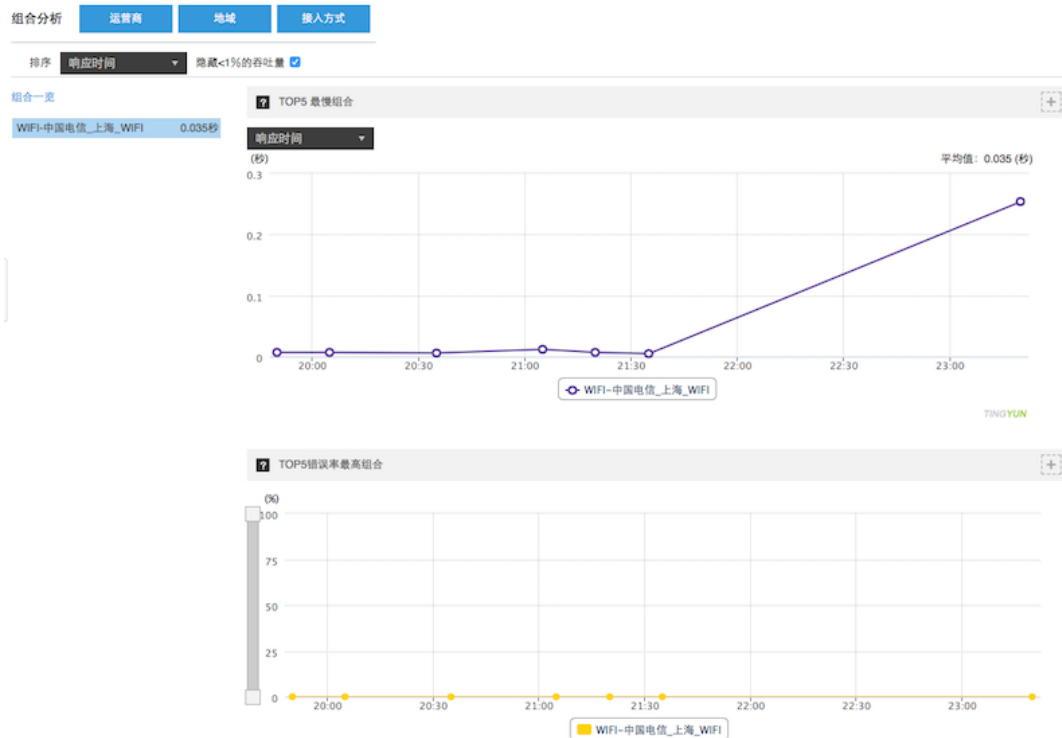
(3) 地域



这个条目比较有意思，用颜色的方式标示出了世界各国的平均响应时间，点击对应的国家还可以继续进入到下一级，最后可以进入到详细的网络数据分析页面。



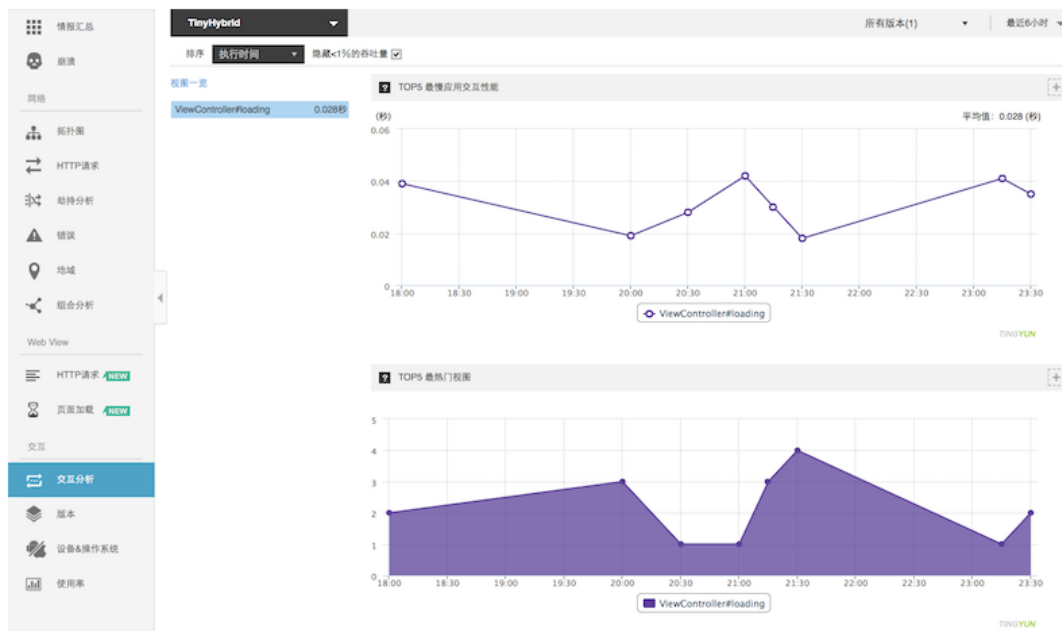
(4) 组合分析



这个页面在中国的网络环境下是非常有用的，它可以根据运营商，地域，接入方式来给出汇总数据。要知道中国的网络环境非常复杂，不同运营商之间，甚至同一运营商在不同的地区网络互通情况会相差非常大。有了这里的数据，就可以有针对性的去部署服务器，优化网络体验。

#### 4. 交互

进入交互分析具体项



在这里我们可以详细的看到 ViewController 以及 View 的每一个系统函数的调用时间，通过这个数据就可以非常好的分析是哪个一个 ViewController 出了问题，对应的去重构就可以了。

交互分析下边的几项是通过一定的条件来看 UI 交互的具体数据，可以通过版本进行过滤，这样就可以方便的过滤掉已经把问题修改掉了的版本。还可以通过操作系统（iOS/Android）和设备来看各自的交互响应的数据。

## 5. 其他

除了性能分析，听云的数据里也有常见的崩溃数据，活跃数以及事件监测等。这里就不详细展开了

## 七 总结

Hybrid App 在某些特定场景是非常有用的，然而也确实有它的局限性，特别是对交互要求很高的地方，使用它是不太合适，毕竟它还是基于网页技术。不过html5在移动端的发展也非常迅速，也许会有更好的未来说不定。总之掌握这个技术是不会错的。另外由于网页端很可能会成为我们性能瓶颈，所以要时时注意测试相关部分的性能表现，也建议使用一些应用性能监测的第三方服务。这样能够更好的定位产品环境的问题。



微信扫一扫

订阅每日移动开发及APP推广热点资讯

公众号：CocoaChina

我要投稿

收藏文章

分享到：

2

上一篇：源码推荐(11.18)：全屏滑动返回，主题模式(一键换肤)

### 相关资讯

iOS 9的 Core Image新滤镜

【译】17个提升iOS开发效率的必备工具

iOS 蓝牙开发（四）BabyBluetooth蓝牙库介绍

这些 iOS 面试基础题目，你都深入了解吗？

[XTPaster] iOS 贴纸功能实现

iOS 开发之 ReactiveCocoa 下的 MVVM（干货分享）

初窥iOS 9联系人框架

iOS 蓝牙开发（三）app作为外设被连接的实现

iOS开发——UI组件（个人整理）

苹果向开发者发布 iOS 9.2 第二个测试版

发现一种创新的开发方式

更智慧的app开发来自IBM

免费试用 IBM Bluemix™ 提升你的app开发效率 →



我来说两句



您还没有登录！请 [登录](#) 或 [注册](#)

所有评论 (1)

wengyajie2015-11-18 05:05:37

以为是干货，后面成广告了

3

0

回复