

QQ空间掉帧率优化实战

原创 2017-12-07 邓荣欣 腾讯WeTest



邓荣欣， 腾讯移动客户端开发工程师

商业转载请联系腾讯WeTest获得授权，非商业转载请注明出处。

WeTest 导读

空间新业务需求日益增多，在业务开发阶段的疏忽，或者是受到其他业务的影响（比如一些非空间的业务网络回包或者逻辑在主线程进行），导致空间的某些页面掉帧率上升。

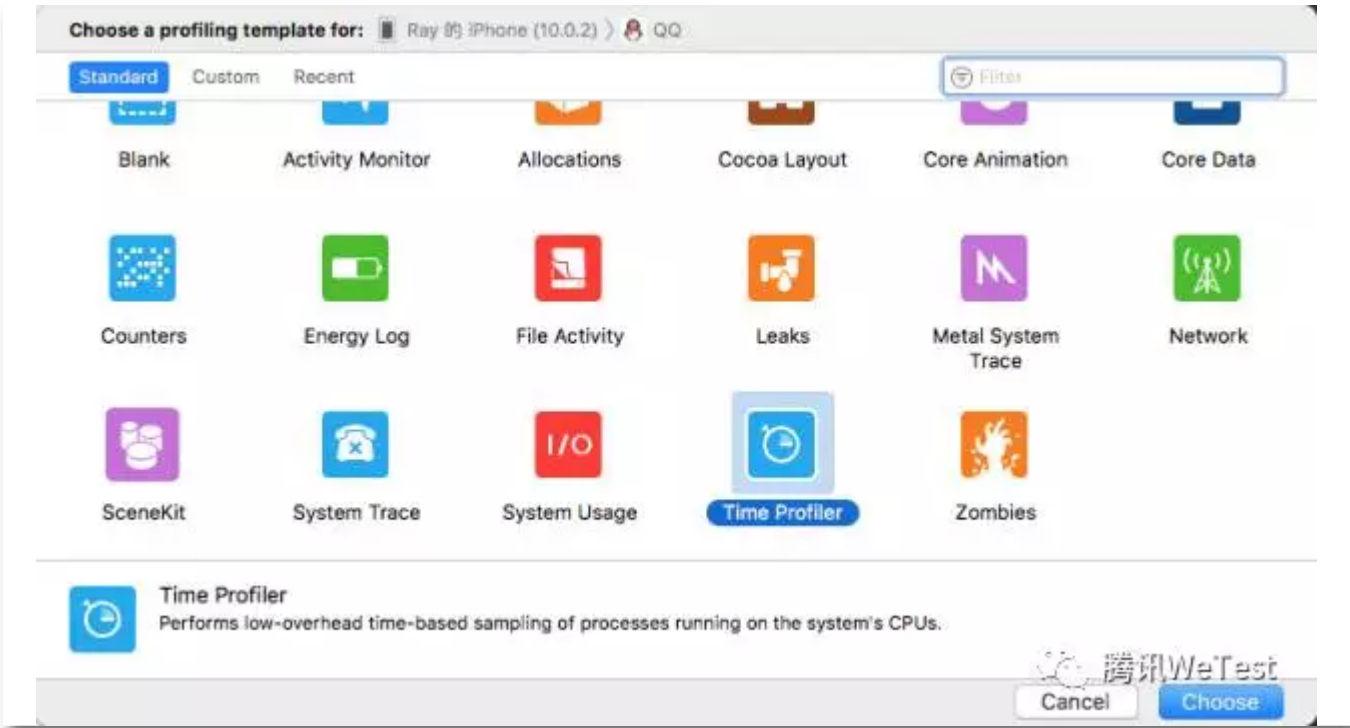
本文从两个方向介绍优化掉帧率：

- Time Profiler时间分析工具
- 一些优化手段

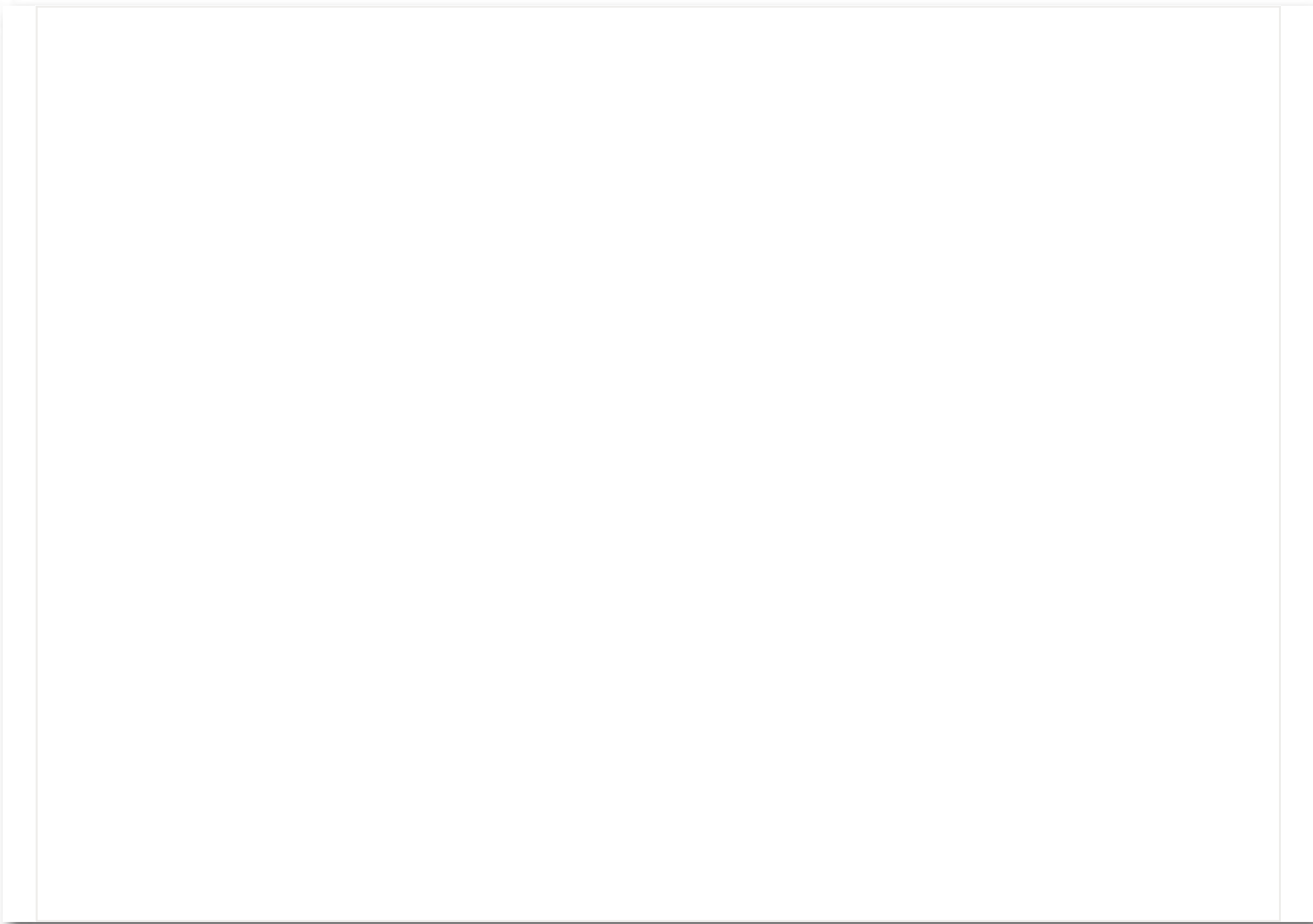
Time Profiler (Xcode 9.1)

“

Time Profiler分析原理：它按照固定的时间间隔来跟踪每一个线程的堆栈信息，通过统计比较时间间隔之间的堆栈状态，来推算某个方法执行了多久，并获得一个近似值。



下面是Time Profiler的界面说明和一些使用建议。



1 主线程使用波峰

开始模拟用户使用App的时候，可以看到主线程的使用情况，它的波峰会忽高忽低，说明app正在进行耗时计算/正常计算，我们可以截取不同时间段的波峰区间进行探究，比如刚进入空间的5秒内，或者拉取到新feeds流之后平缓的5秒等不同场景（小tips：使用触控板向左右两边挪动可以进一步细化时间区间）

2 关于筛选面板的使用

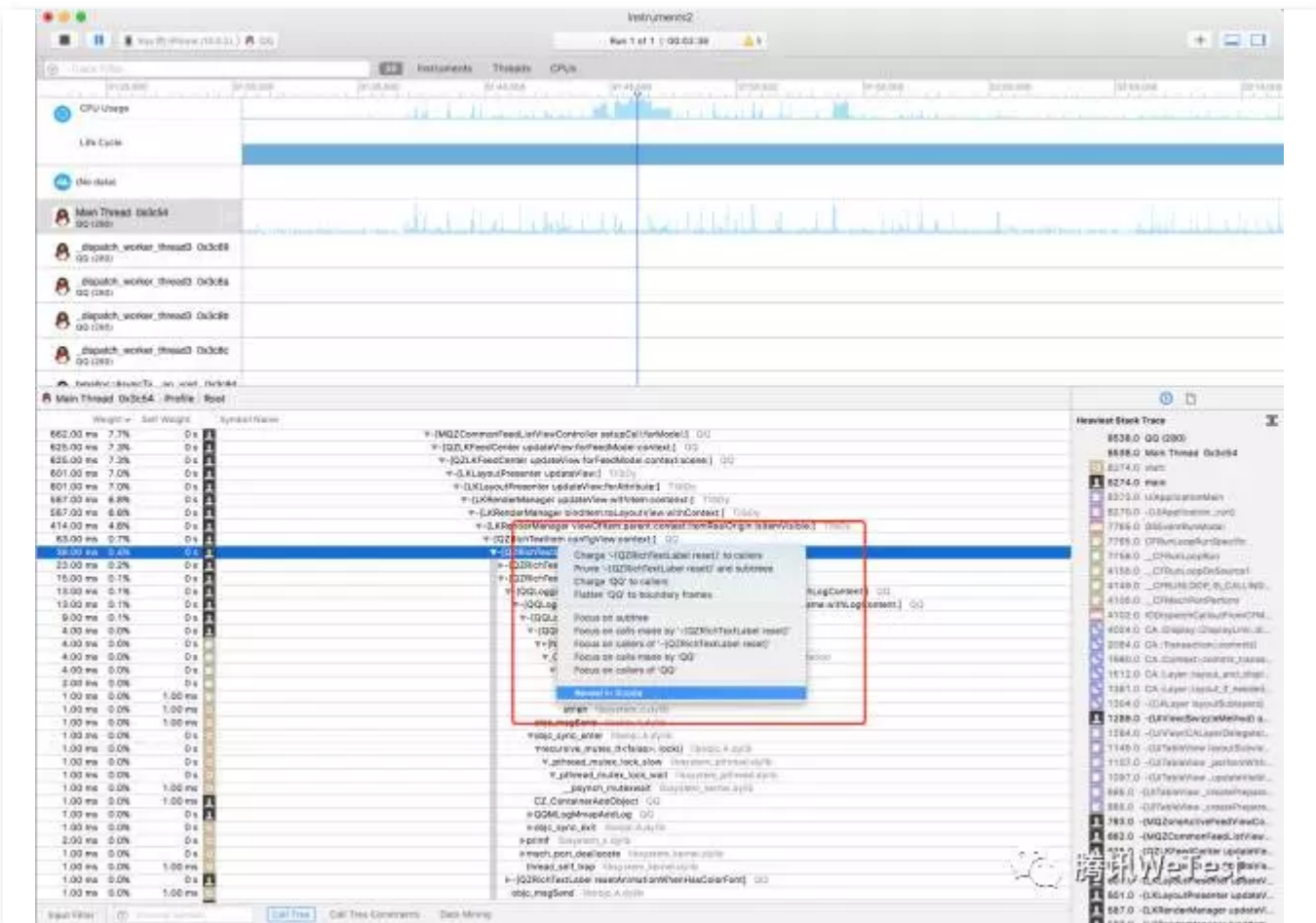
- Separate by State：此选项会根据应用程序的生命周期状态对结果进行分组，并且是检查应用程序在多大程度上执行以及何时执行的有用方法。
- Separate by Thread：根据线程类别分开，方便查看哪些线程占用了最大的CPU。
- Invert Call Tree：调用树倒返过来，将习惯性的从根向下一级一级的显示，如选上就会返过来从最底层调用向一级一级的显示。如果想要查看那个方法调用为最深时使用会更方便些。
- Hide System Libraries：选上它只会展示与应用有关的符号信息，一般情况下我们只关心自己写的代码所需的耗时，而不关心系统库的CPU耗时。（但是很多我们的代码往往是由系统的函数进来，隐藏的话往往可能会丢失很重要的信息）
- Flatten Recursion：将递归函数视为每个堆栈跟踪中的一个条目，而不是多个。

- Top Functions：将花费在函数中的总时间视为直接在该函数内的时间总和以及该函数所调用的函数花费的时间。如果函数A调用B，那么A的时间被报告为A的时间并且加上在B中花费的时间。

为了可以获取更多的信息，建议只勾选 “Separate by Thread”

3 操作小技巧

在折叠的堆栈，按住“alt”点开旁边的三角形即可展开全部折叠堆栈，如果发现耗时严重的堆栈中，可以右键点开菜单，选中“Reveal in Xcodes”即可跳转到对应的代码区域。



实战应用

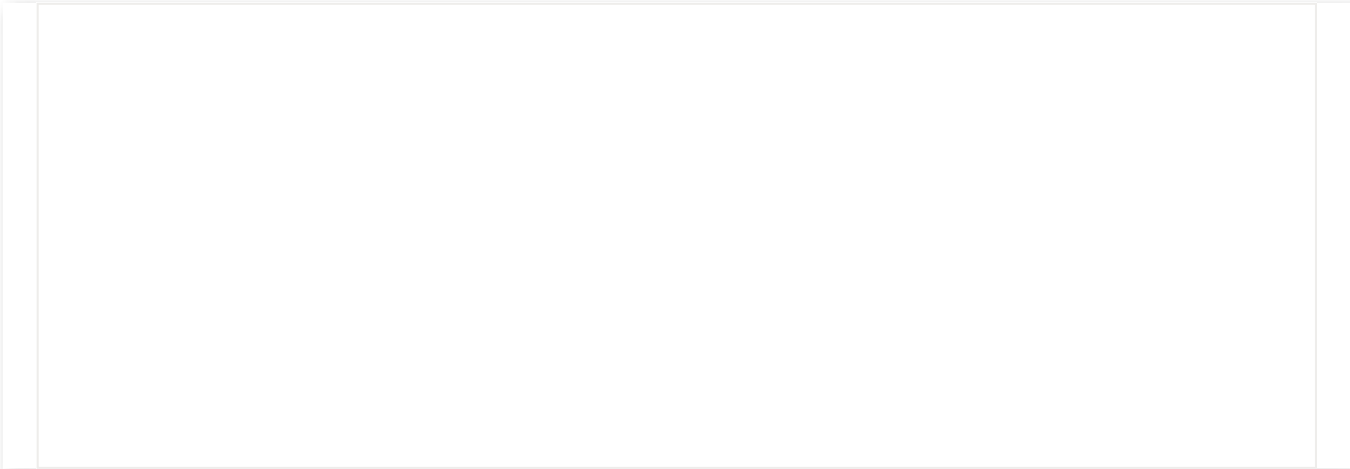
在好友动态页面来回滑动，笔者分四种情况来模拟用户的使用习惯：

- 刚进入空间（无缓存），下拉刷新
- 刚进入空间（有缓存），下拉刷新
- 来回滑动
- 上拉加载更多

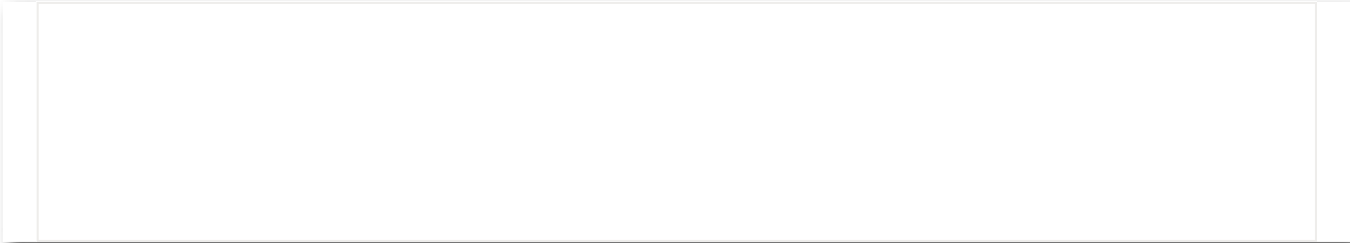
1 将耗时操作（如文件IO）放到工作线程



在我们读取Gif首帧的时候，-[QZoneGIFDecode firstFrameWithURL:viewSize:]里面是有一部是从磁盘里读取二进制文件并且转换成NSData，然后再进行解码，这部分的IO操作优化后是放到了工作线程，异步读取完成解码之后再展示图片，不阻塞主线程。

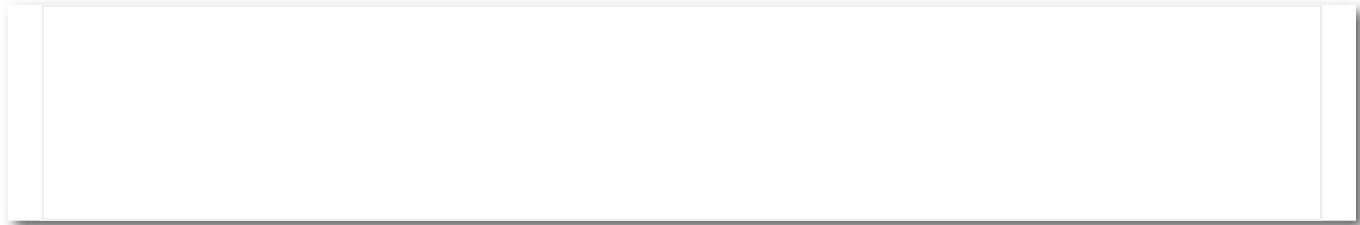


再举一个例子：异步解析网络回包数据和异步排版

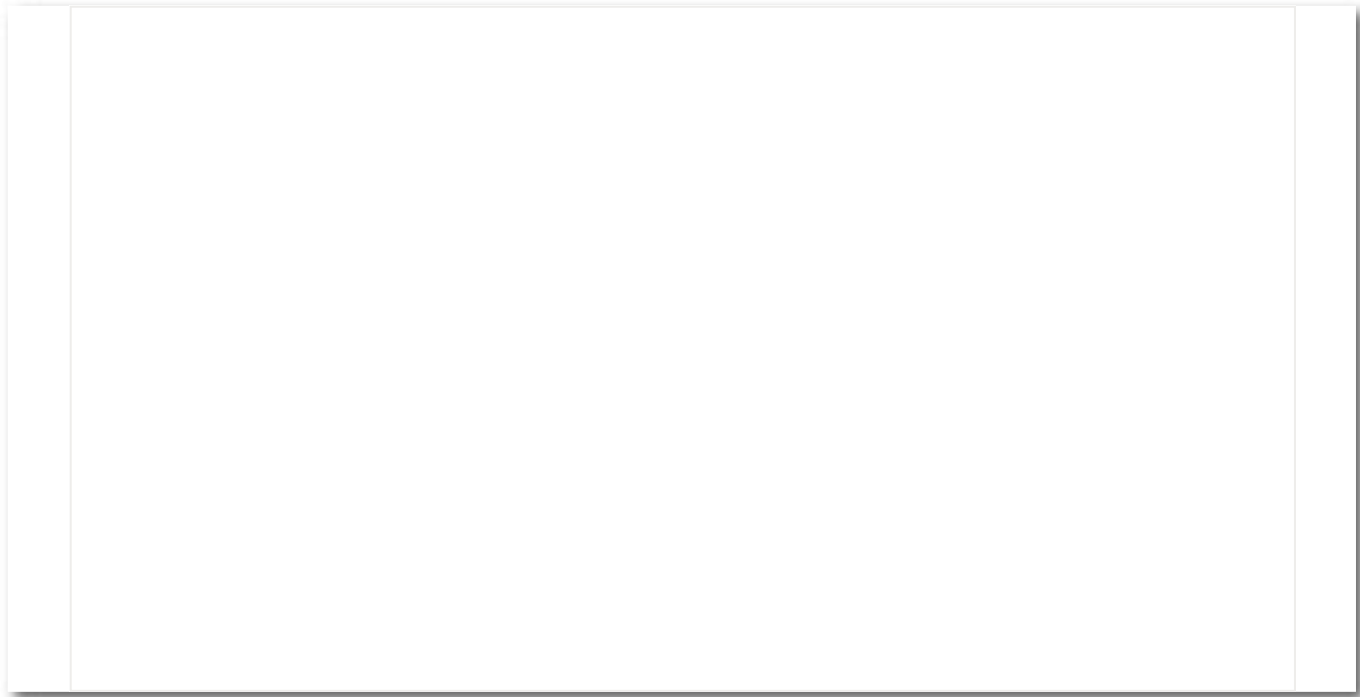


将耗时操作放到工作线程异步执行占了优化工作的大头，在这个过程中，要注意多线程问题，比如线程安全问题、死锁、野指针等，比如容器类的读写操作，最常用的NSMutableArray, NSMutableDictionary等, 这类集合在调用读写方法时并不是线程安全，简单地在里面进行加锁操作是可以保证线程安全，不过也可能会导致其他耗时问题。

2 耗时函数优化



上图堆栈表示的是展示图片， 整个流程如下：

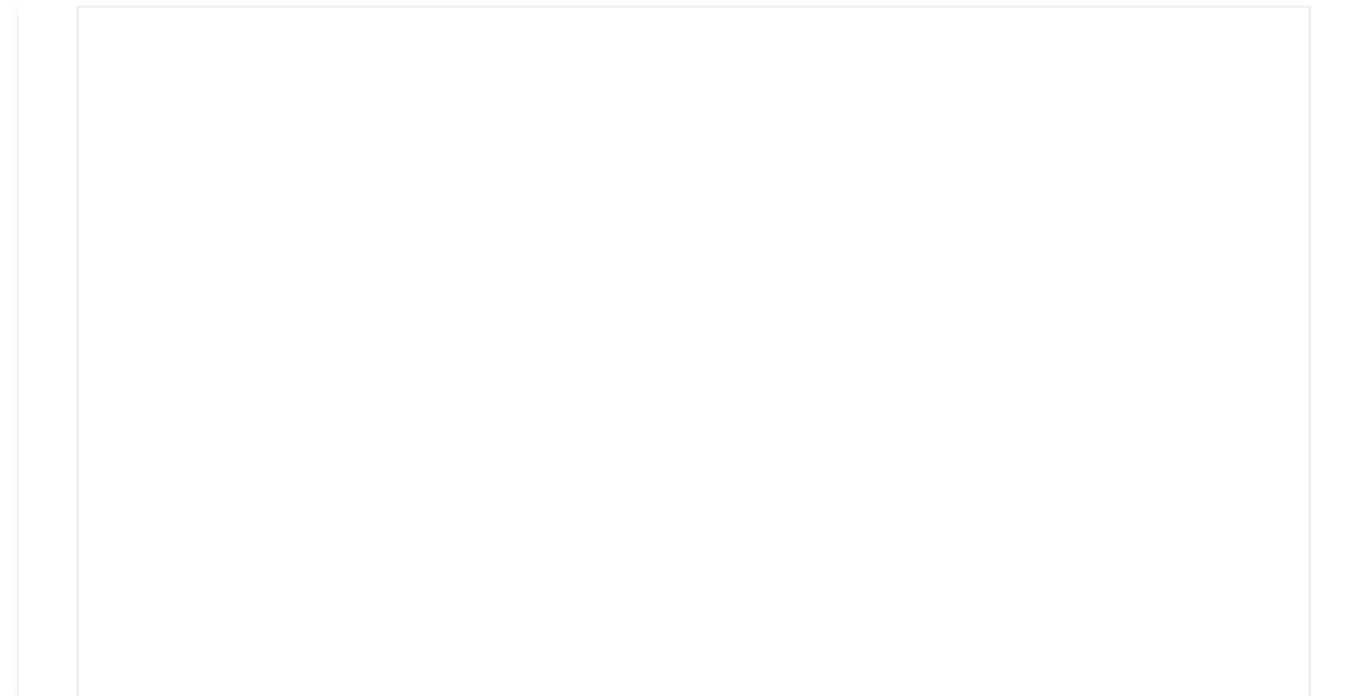


由于空间里面存在大部分图片，其中走网络下载的图片就是上述这个流程。在这个过程中，刨开网络下载的部分，我们会根据图片URL来存取。存取过程首先会将URL 进行MD5加密之后作为Key来进行存取，其实这一步不是必要的，而且系统提供的MD5函数比较耗时。



优化手段：

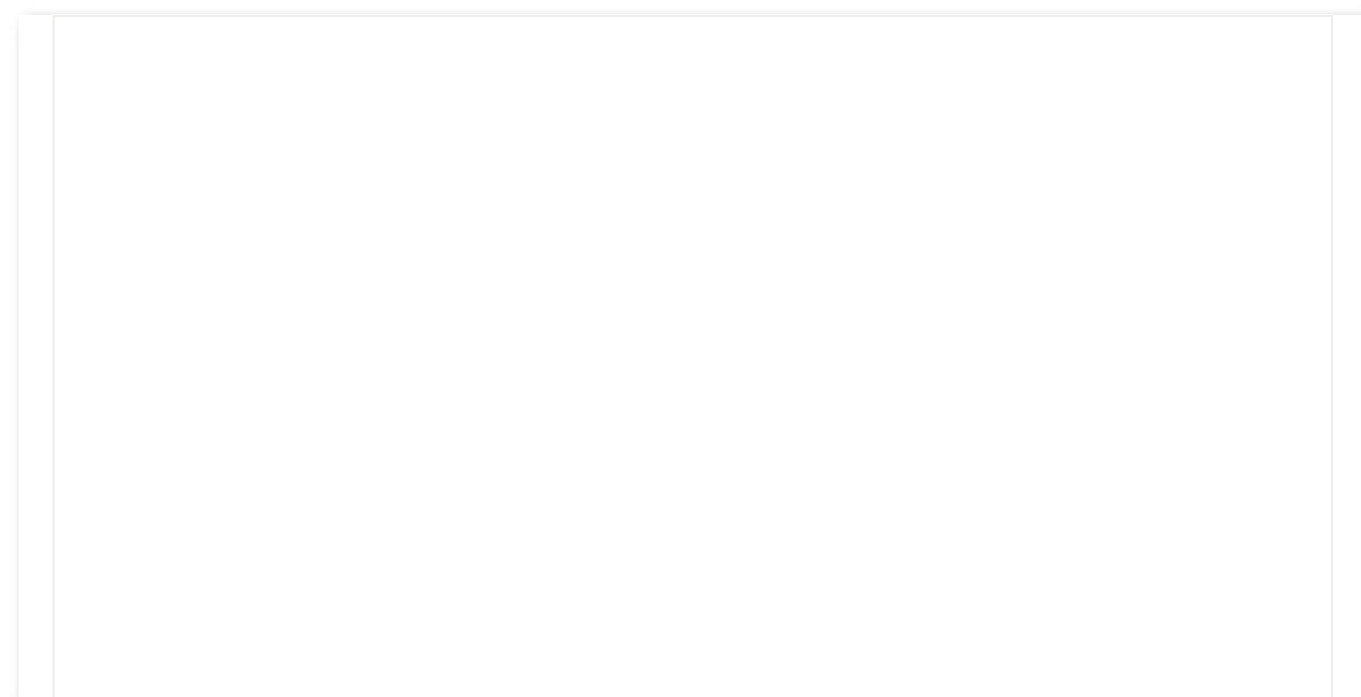
优化缓冲池存取过程，直接使用URL作为Key来存取，去掉MD5调用。



3 减少- (void)scrollViewDidScroll:

(UIScrollView *)scrollView 这个函数里面的耗时操作

这个方法在任何方式触发 contentOffset 变化的时候都会被调用（包括用户拖动，减速过程，直接通过代码设置等），可以用于监控 contentOffset 的变化，并根据当前的 contentOffset 对其他 view 做出随动调整。但是这个方法在滚动的时候每秒调用上百次，如果在里面加入耗时操作就可能对掉帧率造成很大影响。



解决方法：优化调用耗时，或者将耗时操作放到别的地方去

4 提前进行（耗时操作不可避免）

在进入空间之前，我们会有很多初始化工作，比如初始化用户的空间装扮，读取用户的一些配置等，有时候还会涉及IO操作，这部分的耗时是不可避免的。为了保证用户的体验问题，进入空间前，我们可以提前初始化（preload），将一些耗时操作选择在适当的时机提前进行。

5 缓存

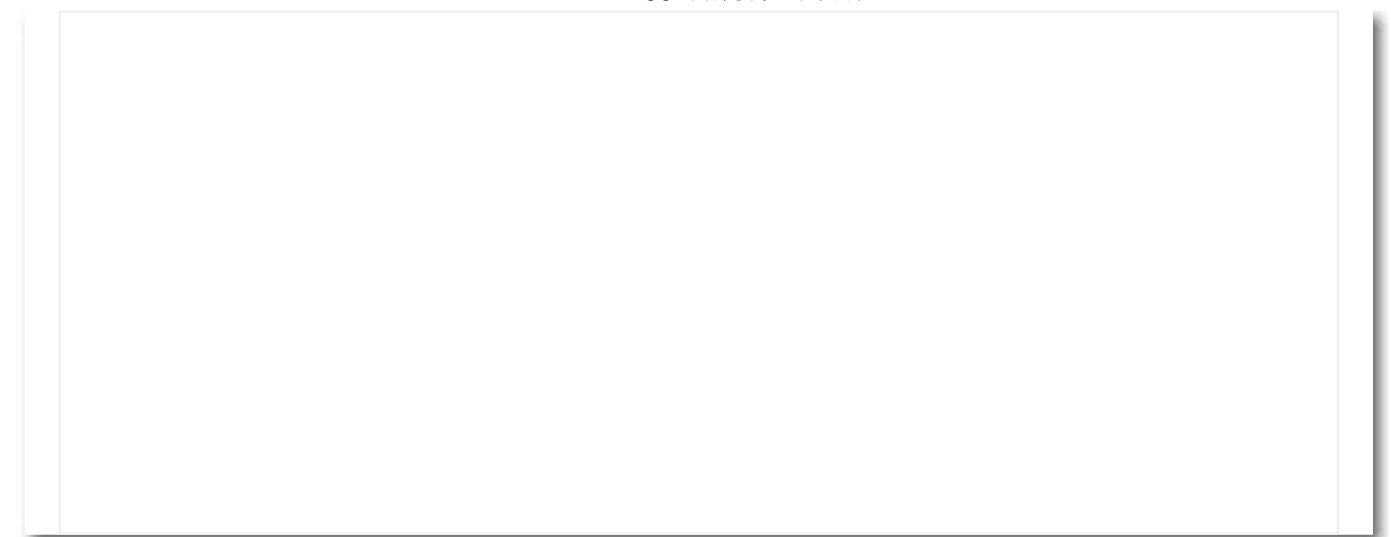
在业务上，我们会读取一些设置项来展示或者进行不同的功能，这些选项的即时读取可能是非常耗时的（尤其是涉及非线性安全容器的读取，里面往往是利用了互斥锁或者信号量等机制保证线程安全，耗时就更加严重），我们可以使用静态变量和dispatch_once来保存起来，避免每次都去要读取一遍。

再举一个例子：

我们在渲染的时候会用到很多字体颜色，每次创建这些新的字体和颜色也是耗时的一部分，我们可以将这些UIColor和UIFont缓存起来，过程如下图：

这里还可以做进一步的优化，就是在进入空间前，把常用的字体生成并且缓存起来，减少渲染时再生成的耗时。

6 减少不必要的操作



为了方便回溯用户的操作行为，我们会在App里面加上很多log，一般log都涉及IO操作，不是必要的log我们要减少，尽量只在关键点打log。

7 懒加载view

不要在cell里面嵌套太多的view，这会很影响滑动的流畅感，而且更多的view也需要花费更多的CPU跟内存。假如由于view太多而导致了滑动不流畅，那就不要在一次就把所有的view都创建出来，把部分view放到需要显示cell的时候再去创建。比如：

```
- (UIButton *)button {
    if (!_button) {
        UIButton *button = [[UIButton alloc] init];
        [self addSubview:button];
        _button = button;
    }
    return _button;
}
```

腾讯WeTest

8 利用主线程不同的runloop

优化缓冲池存取过程，直接使用URL作为Key来存取，去掉MD5调用。

节点

__END__

-[QZWidgetView webView:shouldStartLoadWithRequest:navigationType:]

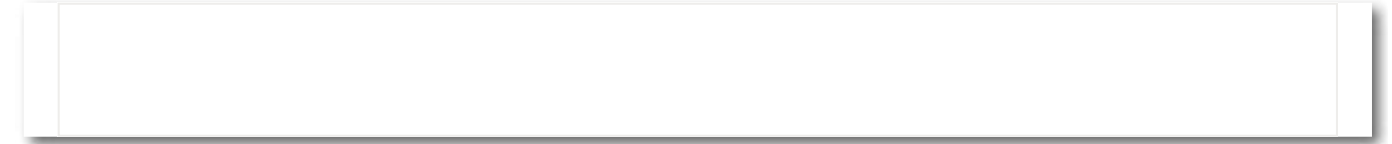
-[QQWebViewController webView:shouldStartLoadWithRequest:navigationType:]

-[NSURLRequest(QzHybridWeb) qzhw_requestKey]

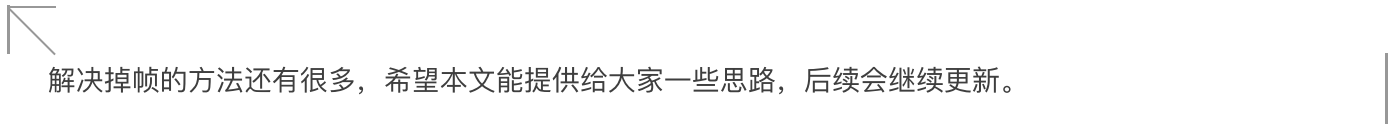
腾讯WeTest

上图是进入空间的时候，需要初始化混合Cover挂件的耗时问题。

我们可以利用不同的runloop来优化这个耗时问题。主线程的 **RunLoop** 里有两个预置的 **Mode**：
kCFRunLoopDefaultMode 和 UITrackingRunLoopMode。这两个 Mode 都被标记为"Common"属性。
DefaultMode 是 App 平时所处的状态，TrackingRunLoopMode 是追踪 ScrollView 滑动时的状态。当我们初始化挂件并加到 DefaultMode 时，这个事件 会得到重复回调，但此时滑动一个TableView时，RunLoop 会将 mode 切换为 TrackingRunLoopMode，这时 初始化挂件函数就不会被回调，因而也不会影响到滑动操作。



很多App会用到这个点来进行优化，比如我们操作微信的时候，在朋友圈上拉加载更多的时候，如果不松开手是不会加载出来的，只有放开手才会展示出更多的feeds，也是利用了不同的runloop。



解决掉帧的方法还有很多，希望本文能提供给大家一些思路，后续会继续更新。

UPA—— 一款针对Unity游戏/产品的深度性能分析工具，由腾讯WeTest和unity官方共同研发打造，可以帮助游戏开发者快速定位性能问题。旨在为游戏开发者提供更完善的手游性能解决方案，同时与开发环节形成闭环，保障游戏品质。

目前，限时内测正在开放中，即日起至2017.12.21，所有预约成功的WeTest平台认证用户，均可以**免费、不限次数**地使用最完整的UPA服务，点击“[阅读原文](#)”立即预约。

对UPA感兴趣的开发者，欢迎加入QQ群：633065352

如果对使用当中有任何疑问，欢迎联系腾讯WeTest企业QQ：800024531

点击左下角“[阅读原文](#)”预约UPA服务

★如果你喜欢这篇文章，欢迎分享到朋友圈★

关于腾讯WeTest

腾讯WeTest是腾讯游戏官方推出的一站式游戏测试平台，用十年腾讯游戏测试经验帮助广大开发者对游戏开发全生命周期进行质量保障。

腾讯WeTest提供：兼容适配测试；云端真机调试；安全测试；耗电量测试；服务器压力测试；舆情监控等服务。



[阅读原文](#)