WILEY

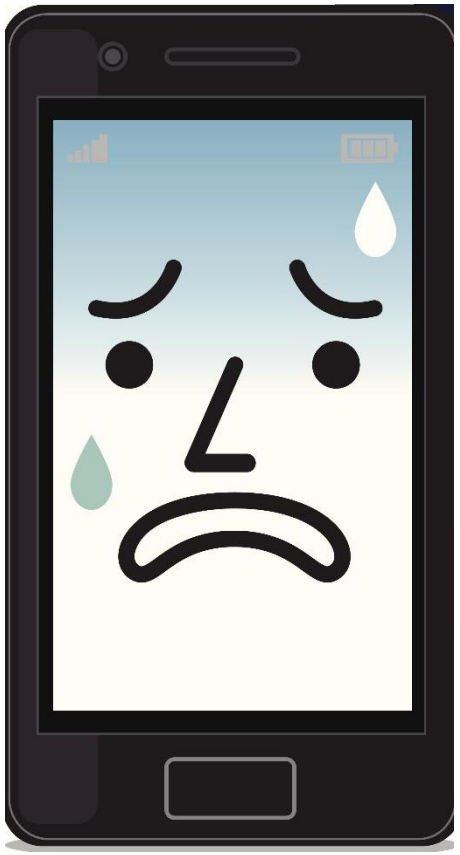
应用的技术分析

营销和推广中付出的金钱和精力

- 应用崩溃
- 阻塞主线程
- 内存泄露
- 使用同步HTTP请求
- 耗流量
- 耗电
- 糟糕用户界面


对应用进行技术分析

应用崩溃



崩溃处理器：

- 为用户显示一条关于崩溃的消息
- 从崩溃日志中收集信息

阻塞主线程

```
dispatch_queue_t bgQueue =
dispatch_get_global_queue
(DISPATCH_QUEUE_PRIORITY_BACKGROUND,
0);
dispatch_async(bgQueue, ^{
//perform your operation
});
```

- 不要在应用的主线程上执行长时间运行的繁重任务
- 将繁重操作移到后台队列，避免同主线程发生干扰

内存泄露

[NSSting [alloc] init] \\ retain count 1

[retain] \\ retain count 2
[retain] \\ retain count 3
.
.
.
[release] \\ retain count 1

内存泄露发生在应用不能正确管理内存分配且内存分配和释放没有得到平衡的时候。

内存泄露

[NSSting [alloc] init] \\ retain count 1

[retain] \\ retain count 2
[retain] \\ retain count 3
.
.
.
[release] \\ retain count 1
[release] \\ retain count 0

内存泄露可以通过自动引用计数（ARC）来解决

ARC可以通过-fobjc-arc明确启用

ARC可以通过-fno-objc-arc明确禁用

使用同步HTTP请求

- NSURLRequest
- NSURLConnection

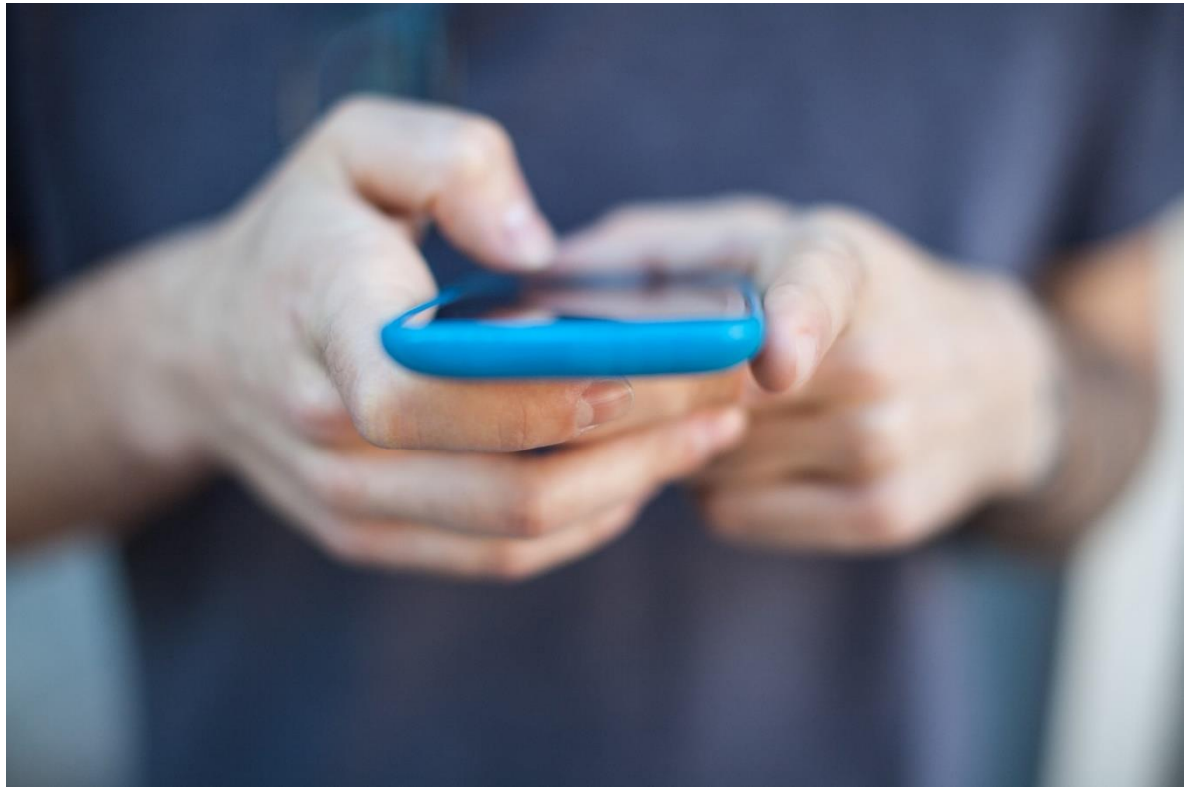
使用同步HTTP请求

- NSURLRequest对象表示一个URL加载请求，独立于协议和URL scheme
- NSURLConnection对象被用于执行NSURLRequest的加载
- 苹果开发者指南很不建议使用同步连接

使用同步HTTP请求

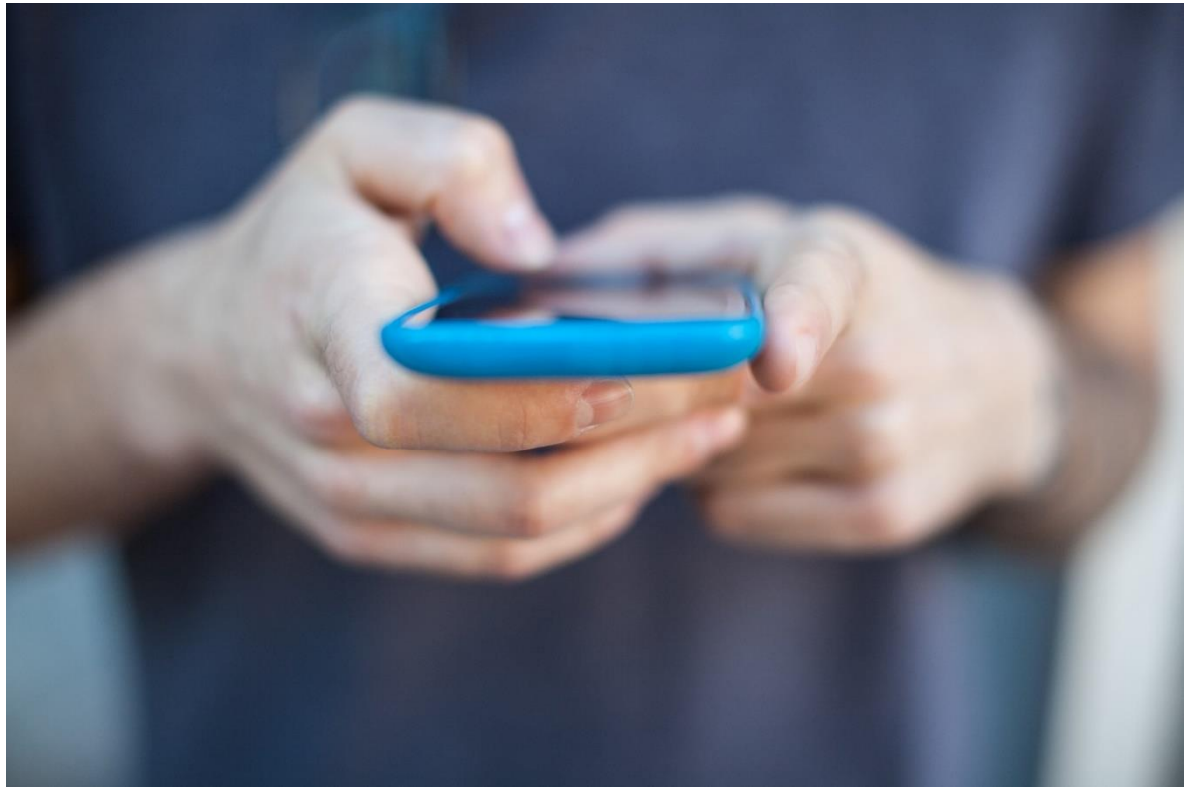
为网络通信使用**URLRequest** 类

耗流量



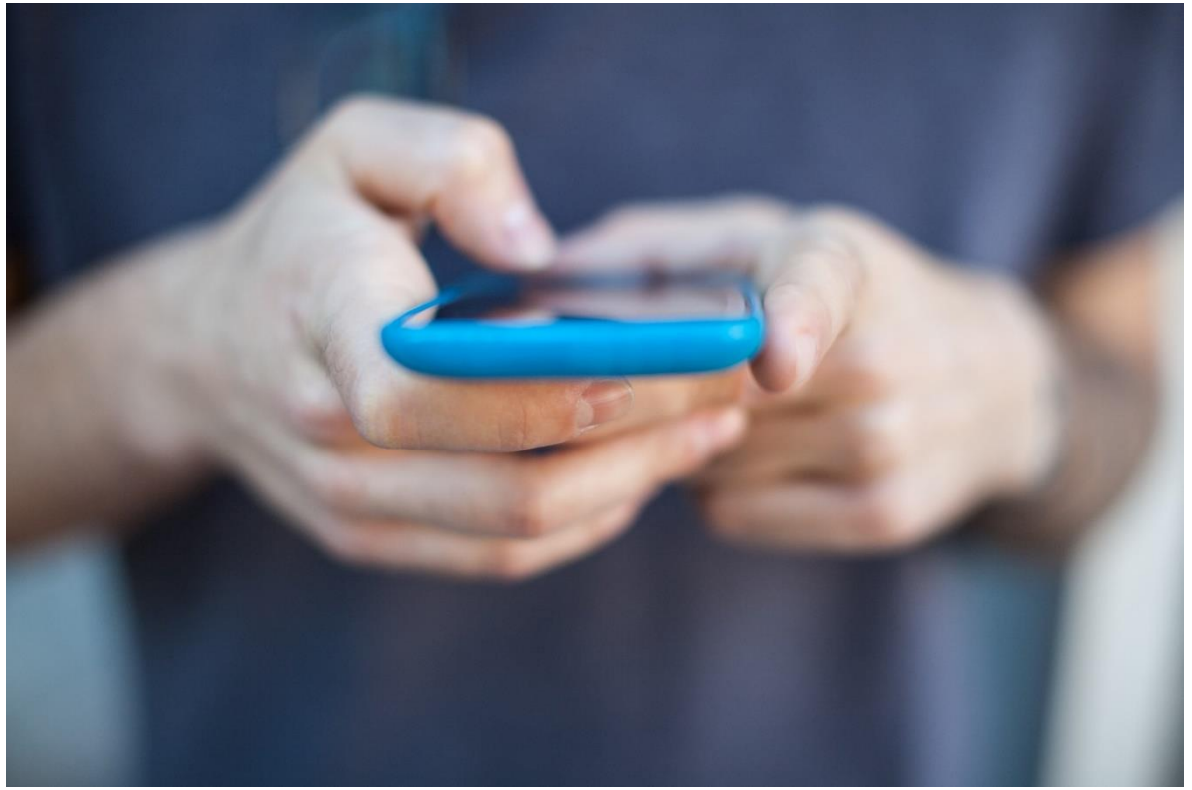
耗流量可能会导致用户向手机运营商支付额外费用

耗流量



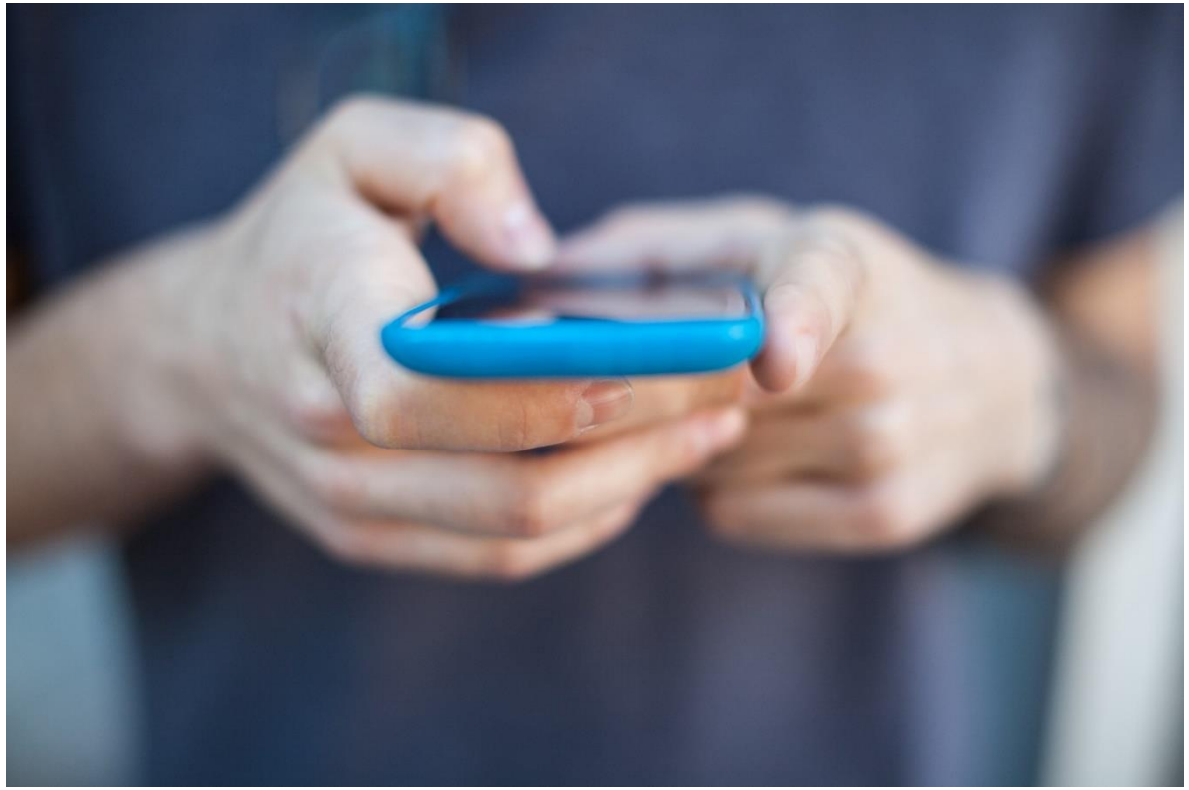
当你的应用同Web服务通信时，不要每次应用启动都重载所有数据集

耗流量



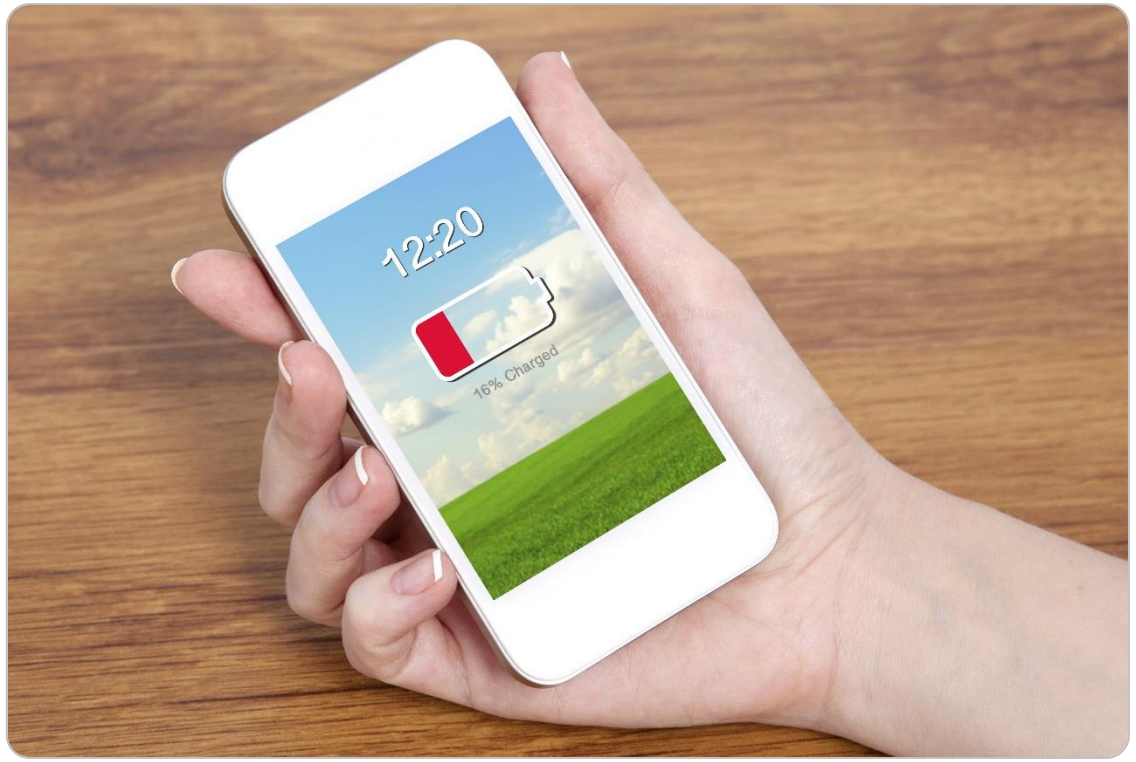
- 检查用户设备连的是Wi-Fi还是3G、4G等手机网络
- 大多数情况下，Wi-Fi都是免费的，而手机网络是和运营商套餐挂钩的

耗流量



使用Reachability来监控设备的网络状态

耗电



一大很耗电的原因是调用了CLLocationManager类的startUpdatingLocation方法而不调用stopUpdatingLocation方法

耗电

- 本地化你的应用
- 创建并初始化CLLocationManager类的实例，设置委托，并在application:didFinishWithLaunchingOptions:方法中调用startUpdateLocation方法
- 实现locationManager:didUpdateToLocations:委托方法并在CLLocation属性中存储[locations lastObject]值

耗电

```
#import <UIKit/UIKit.h>

#import "Reachability.h"

#import <CoreLocation/CoreLocation.h>

@class YDViewController;

@interface YDAppDelegate : UIResponder
<UIApplicationDelegate,CLLocationManagerDelegate
>

@property (strong, nonatomic) UIWindow *window;

@property (strong, nonatomic) YDViewController
*viewController;
```

```
@property(nonatomic,strong) CLLocationManager*
locmanager;

@property(nonatomic,strong) CLLocation
*userlocation;

 @property (strong, nonatomic) Reachability*
reachability;

-(BOOL)connectedViaWIFI;

@end
```

耗电

```
#import "YDAppDelegate.h"

#import "YDViewController.h"

@implementation YDAppDelegate

- (BOOL)application:(UIApplication *)application

didFinishLaunchingWithOptions:(NSDictionary
*)launchOptions
```

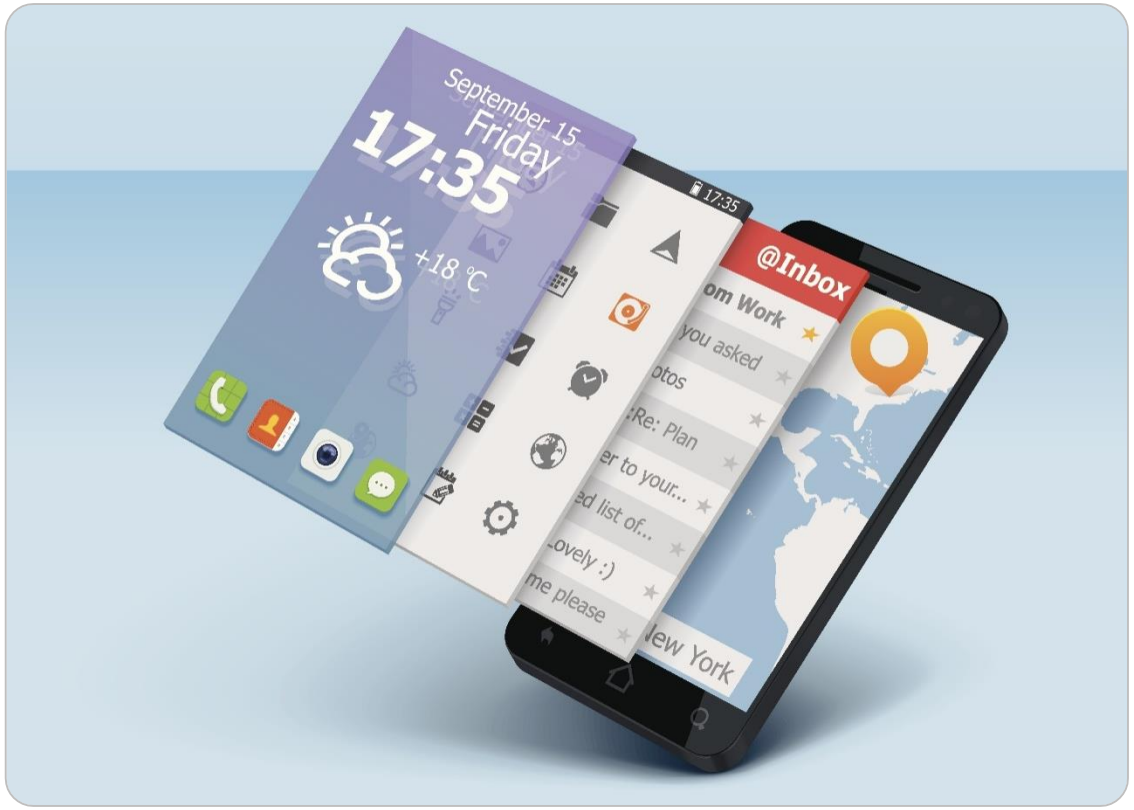
```
{

self.window = [[UIWindow alloc]
initWithFrame:[[UIScreen mainScreen] bounds]];

self.locmanager = [[CLLocationManager alloc] init];

self.locmanager.delegate=self;

[self.locmanager startUpdatingLocation];

self.reachability = [Reachability
reachabilityForInternetConnection];

[self.reachability startNotifier];

[self.reachability currentReachabilityStatus];
```

耗电

```
// Override point for customization after
application launch.

self.viewController = [[YDViewController alloc]

initWithNibName:@"YDViewController" bundle:nil];

self.window.rootViewController =
self.viewController;

[self.window makeKeyAndVisible];

return YES;

}

-(BOOL)connectedViaWIFI

{
```

```
return [self.reachability
currentReachabilityStatus]== ReachableViaWiFi;

}

#pragma mark CoreLocation

- (void)locationManager:(CLLocationManager
*)manager

didUpdateLocations:(NSArray *)locations

{

self.userlocation=[locations lastObject];

[self.locmanager stopUpdatingLocation];

}
```

糟糕用户界面



- 为用户的设备屏幕设计应用
- 使用苹果文档的iOS人机界面指南来设计专业用户界面
- 使用高质量艺术作品来丰富你的应用

怎样在应用中避免流量消耗过大？

a) 使用崩溃处理器，从崩溃日志中收集信息

b) 避免阻塞主线程，将操作移往后台队列

c) 避免每次在用户启动应用时都重新加载整个数据集

d) 使用ARC来管理内存

怎样在应用中避免流量消耗过大？

a) 使用崩溃处理器，从崩溃日志中收集信息

b) 避免阻塞主线程，将操作移往后台队列

c) 避免每次在用户启动应用时都重新加载整个数据集

d) 使用ARC来管理内存

总结

耗流量– 解决办法：

- 避免总是重新加载全部数据
- 检查用户设备连的是Wi-Fi还是3G、4G等手机网络
- 使用Reachability来监控设备的网络状态

耗电 – 解决办法：调用CLLocationManager类的stopUpdatingLocation方法

糟糕用户界面 – 解决办法：

- 为用户的设备屏幕设计应用
- 遵循iOS人机界面指南
- 使用图形设计工具来创建专业的用户界面

WILEY