列表滑动卡顿问题

1. 概述:

项目开发中,免不了和列表打交道,几乎每个页面都含有列表。列表展示的 数据也是多种多样的,各种图片文字各种布局。经常伴随着列表的使用出现的问题就是:列表滑动卡顿现象。有时候会羡慕其他平台开发的 app 列表滑动不会卡,倒 把卡顿问题归咎于"ios 系统不给力"。其实,最主要的原因还是在我们开发自己 代码问题。

2. 列表滑动卡顿问题分析

为什么会卡顿呢?很简单,主线程同步操作过多,延迟了滑动过程中 cell 的展示。所以,遇到滑动列表卡顿这种问题,应该先从列表的几个方法里查找,看 是否存在过多同步操作。主要列表同步操作集中在cellForRow,heightForRow等方法。为什么?因为这两个方法可能存在一些控件高度计算,而计算文本size是一项 耗时的同步操作,假如在 heightForRow 计算一次 cell 高度后,cellForRow 再次对 相应文本进行计算。那么列表滑动卡顿是在所难免的。。。还有什么同步操作会明 显卡 UI 呢?这可就多了,比如列表图片的展示,假如直接同步请求,那就完蛋;又 比如,动态创建控件,在 cellForRow 里反复执行 removeFromSuperView 啊 addSubview 啥的,那么也会造成卡顿,因为控件创建

removeFromSuperView 啊 addSubview 啥的,那么也会造成卡顿,因为控件创建分配内存等操作都是同步操作!

3. 如何解决

这里就没有什么通用的解决方式了,因为同步操作造成主线程阻塞的情况。也说不清,远不止 2 中所提及的情况。主要还是给大家提几个注意事项把,首先,切记 cellForRow 里不要做反复的文本计算,可以把文本计算放在数据请求结果的 json 转 model 数组的过程中,计算好并把高度作为 model 的一个属性,在 cellForRow 和 heightForRow 直接将对应 model 的高度属性值返回即可;再者关于卡顿现象,找到同步代码后,很经常一种解决方式就是多线程处理,那么是不是简单的将同步操作放到子线程就万事大吉了呢?当然不是,因为回到主线程的目的 主要还是为了 cell 的数据展示,那么问题来了,回到主线程对 cell的 UI 进行数据 更新时,此 cell 彼 cell 吗?还是开启子线程时的那个吗?答案是:不一定~具体 原因应该不需要展开了吧,直接上几种解决方案。

方案一: 采用NSOperationQueue

```
NSInvocationOperation *operation = [[NSInvocationOperation alloc]initWithTarget:self
                                            selector:@selector(downloadImage:)
                                              object:indexPath];
 NSOperationQueue *queue = [[NSOperationQueue alloc]init];
 [queue addOperation:operation];
-(void)downloadImage:(NSIndexPath *)indexPath{
  @autoreleasepool {
    // 2.计算字符串占据空间大小
    NSString *stringData=[NSString stringWithFormat:@"<style>*{font-size:16px;}</style>%@",[_DataArray[indexPath.row] valueForKey:@"message"]];
    NSAttributedString * attrStr = [[NSAttributedString alloc] initWithData:[stringData dataUsingEncoding:NSUnicodeStringEncoding] options:@{
      NSDocumentTypeDocumentAttribute: NSHTMLTextDocumentType } documentAttributes:nil error:nil];
    NSMutableDictionary *dic= [[NSMutableDictionary alloc]init];
    [dic setObject:attrStr forKey:@"attrStr"];
     [dic setObject:indexPath forKey:@"indexPath"];
    [[NSOperationQueue mainQueue] addOperationWithBlock:^{
      OtherTableViewCell *cell = [_tableView cellForRowAtIndexPath:[dic objectForKey:@"indexPath"]];
      cell.msgLabel.attributedText= [dic objectForKey:@"attrStr"];
           NSLog(@"%ld %d %@",index.row,b,cell);
    }]:
  }
方案二: 采用GCD

    - (void)loadCenterViewWithHtmlString:(NSString *)htmlStr imageUrl:(NSString *)imageUrl{

    _weak typeof(self) weakSelf = self;
    dispatch_async(dispatch_get_global_queue(DISPATCH_QUEUE_PRIORITY_DEFAULT, 0), ^{
      @autoreleasepool {
        NSIndexPath *indexpath = weakSelf.indexPath;
        NSString *stringData=[NSString withFormat:@"<style>*{font-size:15px;}</style>%@",htmlStr];
        NSAttributedString *str = [[NSAttributedString alloc] initWithData:[stringData dataUsingEncoding:NSUnicodeStringEncoding] options:@{
          NSDocumentTypeDocumentAttribute: NSHTMLTextDocumentType} documentAttributes:nil error:nil];
        dispatch_async(dispatch_get_main_queue(), ^{
          if(indexpath.row == weakSelf.indexPath.row){
            [weakSelf.contentView setAttributedText:str];
```

方案三:数据请求回调中在子线程计算完后回到主线程进行回调及列表刷新 个人更倾向于方案三,因为前两种可能造成列表滑动过程中短暂性空白或短暂性 数据不一致。

4. 关于列表卡顿几个问题及解答

});
});

(1) 预估高度能提高效率?用了会莫名蹿动? 答:对列表熟悉的都知道有一个estimate代理方法,返回预估高度,这种方法可以改变列表数据加载的顺序及一些方法调用频率,比如heightforrow和cellforrow调用频率减少了,不像默认调好多次heightforrow再调cellforrow;但是,这个方法仅在cell高度不变情况下才可作为列表卡顿优化方案,否则反而引起莫名蹿动

- (2) 类似日期格式化等等后台返回的不友好的数据的处理,放cellforrow里正确吗?
 - 答:不正确,应该有相应的模型处理成友好的,cellforrow里直接显示
- (3) cell某个父视图样式很多,所以一股脑将所有涉及子控件在cellforrow里 反复动态创建,对吗?
 - 答:不正确,杜绝控件动态加载
- (4) 关于cell高度,在cellforrow里算一次,heightforrow里再算次,对吗? 答:不正确,尽量只算一次,滑动不算
- (5) 纯代码、storyboard、xib下cell重用方式,都会了吗? 答:必须都会!
- (6) cell,用到富文本来显示html内容,在cellforrow里创建,对吗? 答:注意了,富文本的alloc很耗时而富文本的计算前提又需要富文本对象, 所以cellforow里尽量别重复创建富文本,能懒加载尽量懒加载,想想,怎 么做?
- (7) cellforrow里实在没办法,很多耗时操作吗? 考虑3.中的三种方案试试看吧