

关于贵金属 iOS 客户端圈子详情页卡顿问题的解决过程

问题描述:

用户浏览圈子详情页(如下图), 当文字内容较多时, 会明显的出现卡顿的现象, 影响用户体验.

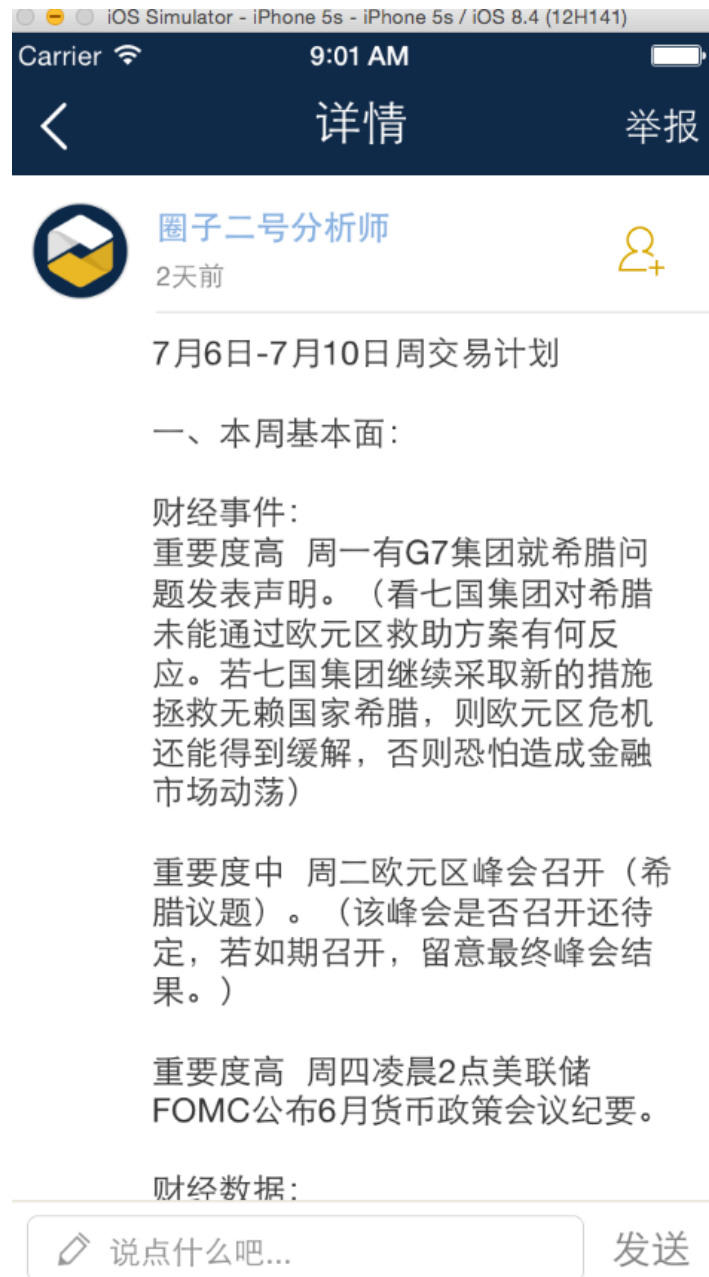


图 1 圈子详情页

问题查找过程:

1. 这个页面是用一个 `tableView` 实现的, 而 `tableView` 继承于 `scrollView`, 而这个页面也正巧也实现了 `UIScrollViewDelegate` 的 `scrollViewDidScroll:` 这个方法, 所以尝试从这个方法入手. 后来发现虽然页面一滑动就会走这个代理方法, 但是将其全部删掉也还是会出现卡顿现象, 故排除是这个方法的原因.

2. 由于这个页面还增加里一个 UITapGestureRecognizer 手势, 所以考虑有可能是它引起的, 遂将其全部注掉, 故障依旧, 发现也不是这个原因.
3. 最后考虑还是 tableView 的数据源代理是不是在一开始加载这个页面的时候做了一些特别的操作, 导致用户滑动页面的时候会触发哪些复杂的操作(虽然数据源方法不会在页面稍有滑动时就会调).

采用的方法依旧是把现实文字的这个 cell 内容分别注释掉. 把整个帖子内容 cell 注释掉, 页面无内容, 也没有卡顿现象. 遂定位肯定是这个 cell 的创建导致卡顿的问题. 接着分别把创建这个 cell 所做的附加操作(如下图), 一一注释掉, 发现只要执行第一行就会复现问题, 继续看这个函数都做了什么.

```
[contentCell setCellByPostBO:self.postBO];
contentCell.clipsToBounds = YES;
contentCell.delegate = self;
```

图 2 圈子详情"tableView:cellForRowAtIndexPath:"

在"LDCPCirclePostDetailContentCell"这个类里的"setCellByPostBO:"这个函数里, 分块注释功能, 发现只要执行了如下处理就会复现现象, 原因是执行了第四行的"addLinkWithTextCheckingResult"这句, 而这个函数是引用 pods 里"TTTAttributedLabel"这个第三方库.

```
//对@进行处理
if (postBO.atInfoArray.count > 0) {
    for (NSDictionary *dic in postBO.atInfoArray) {
        NSValue *value = [dic objectForKey:CPCircleKeyAtRange];
        [self.contentLabel addLinkWithTextCheckingResult:[NSTextCheckingResult transitInformationCheckingRes
            dictionaryWithObject:[dic objectForKey:CPCircleKeyLinkInfo] forKey:NSTextCheckingAirlineKey]]];
    }
}
```

图 3 显示帖子的 cell 中"setCellByPostBO:"

所以到目前为止, 可能因为 2 个地方导致问题的出现: (1) 帖子中并没有"@"等特殊符号, 为什么 postBO.atInfoArray 里面为什么会有东西(而且调试发现, 还是个空内容); (2) TTTAttributedLabel 这个库到底执行了哪些复杂的操作导致 cell 里的 label 会如此卡.

先顺着注释代码的思路继续跟"TTTAttributedLabel"源码, 找上面提的第二个疑问. 最后跟到如下这个函数时发现帖子内容只要一被拖动, 这个函数就回反复被执行, 根据 github 上这个库的介绍, 和代码阅读, 猜想这个函数就根据用户点击的点来反复查找 label 里是否有可以跳转的内容, 如"超链接, 电话号码等". 最后一句一堆"?"语句更是会一点一点的扩大搜索半径进行查找, 这势必会耗费大量的时间, 将搜索边境固定为 2.5 个 f, 卡顿现象大有改善, 已经在可以接受的范围内了.

```
- (NSTextCheckingResult *)linkAtPoint:(CGPoint)point {
    // Stop quickly if none of the points to be tested are in the bounds.
    if (!CGRectContainsPoint(CGRectInset(self.bounds, -15.f, -15.f), point) || self.links.count == 0) {
        return nil;
    }

    // Approximates the behavior of UIWebView which will trigger for links on touches within 15pt of the edge.
    return [self linkAtIndex:[self characterIndexAtPoint:point]]
        ?: [self linkAtRadius:2.5f aroundPoint:point]
        ?: [self linkAtRadius:5.f aroundPoint:point]
        ?: [self linkAtRadius:7.5f aroundPoint:point]
        ?: [self linkAtRadius:12.5f aroundPoint:point]
        ?: [self linkAtRadius:15.f aroundPoint:point];
}
```

图 4 "TTTAttributedLabel" 1.13.1 代码 (工程里的版本)

发现这个 pod 在 github 上已有新版本 1.13.4 更新一下 pod 发现上面的代码升级为下图,多了一个开关来控制是否打开往复搜索链接的代码.

```
- (TTTAttributedLabelLink *)linkAtPoint:(CGPoint)point {  
    // Stop quickly if none of the points to be tested are in the bounds.  
    if (!CGRectContainsPoint(CGRectInset(self.bounds, -15.f, -15.f), point) || self.links.count == 0) {  
        return nil;  
    }  
  
    TTTAttributedLabelLink *result = [self linkAtIndex:[self characterIndexAtPoint:point]]  
    ;  
  
    if (!result && self.extendsLinkTouchArea) {  
        result = [self linkAtRadius:2.5f aroundPoint:point]  
        ?: [self linkAtRadius:5.f aroundPoint:point]  
        ?: [self linkAtRadius:7.5f aroundPoint:point]  
        ?: [self linkAtRadius:12.5f aroundPoint:point]  
        ?: [self linkAtRadius:15.f aroundPoint:point];  
    }  
  
    return result;  
}
```

图 5 “TTTAttributedLabel” 1.13.4 代码 (github 上最新代码)

所以上面提的第二个疑问“TTTAttributedLabel 这个库里到底执行了哪些复杂的操作导致 cell 里的 label 会如此卡”已经解决并找到解决办法: 设置 `self.extendsLinkTouchArea = NO`; 改为如图 6 所示.

```
//对@进行处理  
if (postBO.atInfoArray.count > 0) {  
    for (NSDictionary *dic in postBO.atInfoArray) {  
        NSValue *value = [dic objectForKey:CPCircleKeyAtRange];  
        self.contentLabel.extendsLinkTouchArea = NO; //edit by wangchao 因改善性能需关闭这个属性  
        [self.contentLabel addLinkWithTextCheckingResult:[NSTextCheckingResult transitInforma  
            dictionaryWithObject:[dic objectForKey:CPCircleKeyLinkInfo] forKey:NSTextCheckingA  
    }  
}
```

图 6 疑问 2 的解决方案(比图 3 多了一行)

接下来继续查找问题(1)“帖子中并没有”@“等特殊符号, 为什么 postBO.atInfoArray 里面为什么会有东西”. 定位到调用 TTTAttributedLabel 的地方, 见图 3. 断点设在那里, 发现 postBO.atInfoArray 里是有一项, 但是里面并没有内容. 继续往上找谁给它赋的值. 于是找到“LDCPCirclePostBO”中的 atInfoArray 的 getter/setter 方法, 发现问题所在, 原代码为图 7.

```

- (void)updateAtInfo
{
    NSString *postContent = _postContent;
    NSError *error = NULL;
    static NSRegularExpression *regex = nil;

    if (regex == nil) {
        regex = [NSRegularExpression regularExpressionWithPattern:@"<@[\s\S]
+?href=\"(.*)\"[\s\S]*?>(.*?)</@>" options:
        NSRegularExpressionCaseInsensitive error:&error];
    }

    NSMutableArray *infoArray = [NSMutableArray array];
    do {
        NSTextCheckingResult *result = [regex firstMatchInString:postContent
        options:NSMatchingReportProgress range:NSMakeRange(0, [postContent
        length])];
        NSString *linkInfo = [postContent substringWithRange:[result
        rangeAtIndex:1]];
        NSString *nickName = [postContent substringWithRange:[result
        rangeAtIndex:2]];
        NSRange *newRange = [NSValue valueWithRange:NSMakeRange([result
        rangeAtIndex:0].location, nickName.length)];
        postContent = [regex stringByReplacingMatchesInString:postContent
        options:NSMatchingReportProgress range:[result rangeAtIndex:0]
        withTemplate:nickName];

        NSDictionary *dic = [NSDictionary dictionaryWithObjectsAndKeys:
        linkInfo, CPCircleKeyLinkInfo, nickName, CPCircleKeyAtNickName,
        newRange, CPCircleKeyAtRange, nil];
        [infoArray addObject:dic];
    } while ([regex numberOfMatchesInString:postContent options:
    NSMatchingReportProgress range:NSMakeRange(0, [postContent length])] !
    = 0);

    _postContent = postContent;
    _atInfoArray = [NSArray arrayWithArray:infoArray];
    _dirty = NO;
}

```

图 7 “LDCPCirclePostBO”里的 atInfoArray setter 方法

代码里执行的是一个 **do while** 循环, 导致第一次执行的时候缺少 **result** 为 **nil** 的判断, 导致没有超链接的帖子也加了一项空的 **postBO.atInfoArray**, 导致后续一系列没必要的复杂操作, 遂修改为图 8。

```

//edit by wangchao 把 do while 循环改为 while do 否则缺失第一次判断,导致没有超链接的帖子也被标记为有超链接,TTTAttributedLabel反复检测用户点击,页面也因此会卡顿。
while ([regex numberOfMatchesInString:postContent options:
    NSMatchingReportProgress range:NSMakeRange(0, [postContent length])] !
    = 0) {
    NSTextCheckingResult *result = [regex firstMatchInString:postContent
        options:NSMakeRange(0, [postContent length])];
    NSString *linkInfo = [postContent substringWithRange:[result
        rangeAtIndex:1]];
    NSString *nickName = [postContent substringWithRange:[result
        rangeAtIndex:2]];
    NSRange *newRange = [NSValue valueWithRange:NSMakeRange([result
        rangeAtIndex:0].location, nickName.length)];
    postContent = [regex stringByReplacingMatchesInString:postContent
        options:NSMakeRange(0, [postContent length])
        withTemplate:nickName];

    NSDictionary *dic = [NSDictionary dictionaryWithObjectsAndKeys:
        linkInfo, CPCircleKeyLinkInfo, nickName, CPCircleKeyAtNickName,
        newRange, CPCircleKeyAtRange, nil];
    [infoArray addObject:dic];
}

```

图8 疑问1的解决方案

至此,2个问题全部解决完毕,图1页面丝毫没有卡顿现象,把解决问题的过程记录下来,与大家分享,欢迎大家指正分析不对的地方.谢谢大家.