TableViewCell与SDWebImage之间的苟且

- 1.实验目的:降低内存消耗,提高滑动效率,提高用户体验度。
- 2.实验IDE: Xcode、Instrument、豆果美食App。
- **3.实验设计**:滑动首页至底部即加载25页数据,利用Instrument中的 TimeProfile工具查看耗时主要矛盾。修改主要矛盾,再重复TimeProfile操作,确定是否解决问题。

4.实验结果:

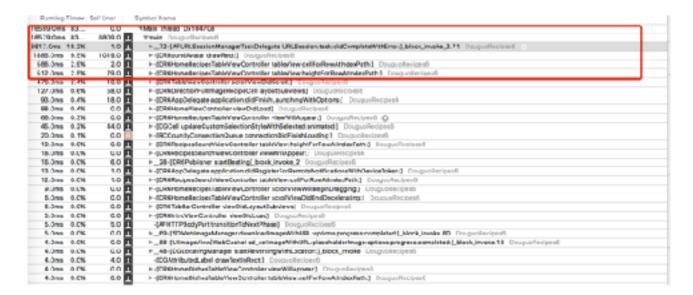
1.利用目前线上的代码查看耗时:



经过代码跟进,发现主要时间消耗是在SDWebImageView对图片做降质操作,对应位置是在SDWebImageDecoder.m文件中的CGContextDrawImage()方法。当然SDWebImage做了对MemoryWarning处理,即删除内存中图片缓存,但经过Instrument查证删除内存图片缓存时,内存会立刻回升,查找后发现每次tableView上拉加载更多时会刷新整个tableView,这样会导致每行都重新设置SDWebImage,当然耗时指向还是SDWebImageDecoder.m文件中的CGContextDrawImage()方法。

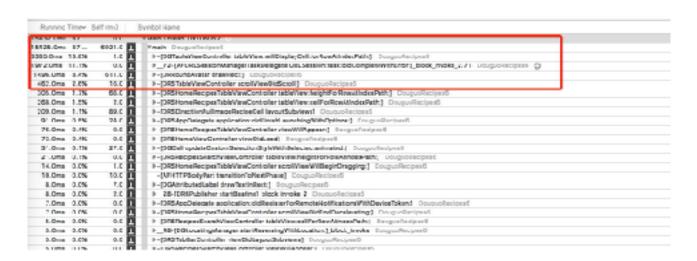
因此提高效率可以从两方面进行,第一:只为tableView显示的Cell设置SDWebImage,这样就会减少很多操作;第二:若服务器下发的图片不太大的话,就不做降质操作。

2.图片不降质



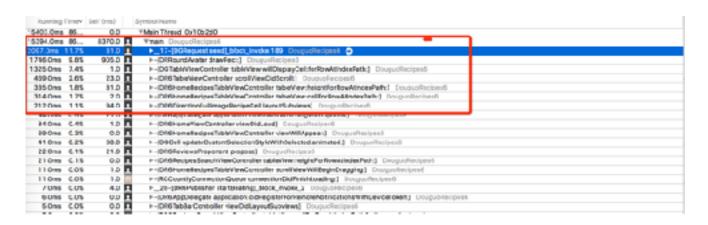
结果:和1操作时间节省了很多

3.只为tableView显示的Cell设置SDWebImage



结果:和1操作时间节省了很多

4.图片不降质量且tableView显示的Cell设置SDWebImage



结果:和1、2、3操作时间节省了

5.个人想法:如果从最小改动+避免修改第三方内部代码,选择第三种方式优化;当然从极致出发采用第四种方法;还有可以优化点,就是在tableView上拉加载数据时,不要执行reloadData,而是insertRows,这样就会减少cell刷新,因为cell中的Label计算高度也是一个耗时项。

6.局限性:由于以上数据都以1次数据,并不太准确,但从代码逻辑上效率:4>2、3>1