

[TD]ContactList scroll , page blank and interact slowly

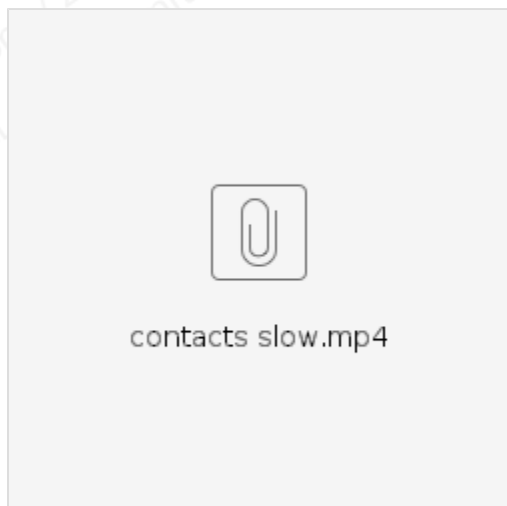
- 1. Project Info
- 2. Background and Purpose
- 3. Analysis
- 4. Improvements
- 5. Improvements Results
- 6. Outlook
- 7. Reference

1. Project Info

| | |
|-------------|--|
| Jira Ticket | <input checked="" type="checkbox"/> SPPAY-77335 - [WS][RN][Tech] optimizing contacts page performance TECH DESIGN |
| dev | guankun.chen@seamoney.com |
| page | Contacts List Page |

2. Background and Purpose

In contact list page, we use Virtualized List to render list, on account of contacts too large, while user scrolling, the page will be blank and interact (liking click event) very slow.



so we aim to avoid page blank and improve interact speed and scroll smoothly

3. Analysis

from the video, we notice the JS FPS very slow almost equal to 0, even if the page initiate to render the list. While scrolling, it sometimes be a negative number.

Based on this, we assume our code is doing massive calculation.

Checking the code, we found renderItem just return a item-component, item-component will be rendered several times while initializing and scrolling. Rendering redundantly is worse, especially the list is large and complex.

```

.....non-memo.....Aaaa 8613512736001
.....non-memo.....Abc 192 8376 79
.....non-memo.....Abcd 00 1 234-567
.....non-memo.....Aaaa 8613512736001
.....non-memo.....Abc 192 8376 79
.....non-memo.....Abcd 00 1 234-567
.....non-memo.....Ccc 159 1407 1249
.....non-memo.....Giao1 180 9178 9055
.....non-memo.....giắc cắm 180 9178 9058
.....non-memo.....Th 6591235556
.....non-memo.....Vn 84987451452
.....non-memo.....Aaaa 8613512736001
.....non-memo.....Abc 192 8376 79
.....non-memo.....Abcd 00 1 234-567
.....non-memo.....Ccc 159 1407 1249
.....non-memo.....Giao1 180 9178 9055
.....non-memo.....giắc cắm 180 9178 9058
.....non-memo.....Th 6591235556
.....non-memo.....Vn 84987451452
.....non-memo.....z p 155 5953 7280
.....non-memo.....+0021692711354 +0021692711354
.....non-memo.....+0021699236252 +0021699236252
.....non-memo.....+00393926714151 +00393926714151
.....non-memo.....+1 203-833-9656 +1 203-833-9656
.....non-memo.....+1 305-342-3557 +1 305-342-3557
.....non-memo.....+1 305-887-5000 +1 305-887-5000
.....non-memo.....+1 310-346-5921 +1 310-346-5921
.....non-memo.....Aaaa 8613512736001
.....non-memo.....Abc 192 8376 79
.....non-memo.....Abcd 00 1 234-567
.....non-memo.....Ccc 159 1407 1249
.....non-memo.....Giao1 180 9178 9055
.....non-memo.....giắc cắm 180 9178 9058
.....non-memo.....Th 6591235556
.....non-memo.....Vn 84987451452
.....non-memo.....z p 155 5953 7280
.....non-memo.....+0021692711354 +0021692711354
.....non-memo.....+0021699236252 +0021699236252
.....non-memo.....+00393926714151 +00393926714151
.....non-memo.....+1 203-833-9656 +1 203-833-9656
.....non-memo.....+1 305-342-3557 +1 305-342-3557
.....non-memo.....+1 305-887-5000 +1 305-887-5000
.....non-memo.....+1 310-346-5921 +1 310-346-5921
.....non-memo.....+1 314-285-4714 +1 314-285-4714
.....non-memo.....+1 318-416-3585 +1 318-416-3585
.....non-memo.....+1 321-310-5318 +1 321-310-5318
.....non-memo.....+1 321-961-0260 +1 321-961-0260
.....non-memo.....+1 323-674-5445 +1 323-674-5445
.....non-memo.....+1 443-802-4935 +1 443-802-4935
.....non-memo.....+1 484-719-1850 +1 484-719-1850
.....non-memo.....+1 619-710-8937 +1 619-710-8937
.....non-memo.....+1 631-507-7324 +1 631-507-7324
.....non-memo.....+1 646-623-9260 +1 646-623-9260
.....non-memo.....Aaaa 8613512736001
.....non-memo.....Abc 192 8376 79
.....non-memo.....Abcd 00 1 234-567
.....non-memo.....Ccc 159 1407 1249
.....non-memo.....Giao1 180 9178 9055
.....non-memo.....giắc cắm 180 9178 9058
.....non-memo.....Th 6591235556
.....non-memo.....Vn 84987451452
.....non-memo.....z p 155 5953 7280
.....non-memo.....+0021692711354 +0021692711354
.....non-memo.....+0021699236252 +0021699236252
.....non-memo.....+00393926714151 +00393926714151
.....non-memo.....+1 203-833-9656 +1 203-833-9656
.....non-memo.....+1 305-342-3557 +1 305-342-3557
.....non-memo.....+1 305-887-5000 +1 305-887-5000

```

while blank page occur, RN will tips like below

```

LOG VirtualizedList: You have a large list that is slow to update - make sure your renderItem function renders components that follow React performance best practices like PureComponent, shouldComponentUpdate, etc. {"contentLength": 5621.0908203125, "dt": 515, "prevDt": 565}

```

4. improvements

| main improvement | detail | minor improvements | detail |
|---------------------------|--|--|---|
| wrap item with React.memo | From the recommendation of official optimizing, we know that using shouldComponentUpdate to avoid list-item render redundantly. While using React Hooks, we should use React.memo to wrap render-item. | Avoid anonymous arrow function on renderItem props | Move out the renderItem function to the outside of render function, so it won't recreate itself each time render function called. |

| | | | |
|--|--|--|--|
| | | Define the key prop on your Item Component | Simply it will avoid re-render on dynamically added /removed items with defined key on each item. Make sure it is unique, don't use index as the key! You can also using keyExtractor as an alternative. |
| | | increase windowSize from 21 to 51 | For a bigger windowSize, you will have more memory consumption. For a lower windowSize, you will have a bigger chance of seeing blank areas. |

5. Improvements Results

after improvements, the JS FPS from almost -2 ~ 2 to almost 10. The initializing and scrolling are smooth

| device | | before | after | remark |
|----------------------|--------|-----------------------|-----------------------|--|
| xiaomi / list 220 | JS FPS | -2 ~ 2 | average 10 and above | scrolling very smoothly |
| iphone 8 / list 1400 | JS FPS | 0 ~ 3 | 15 around | |
| | Memory | cost more around 20MB | cost more around 40MB | because increasing windowSize, but it cost not very much |



6. outlook

Using VirtualizedList cant avoid blank if scrolling too fast, from the official: In order to constrain memory and enable smooth scrolling, content is rendered asynchronously offscreen. This means it's possible to scroll faster than the fill rate and momentarily see blank content. This is a tradeoff that can be adjusted to suit the needs of each application.

Maybe we can investigate [RecycleView from Shopee SDK](#) that using native to solve problem of LargeList

7. reference

- <https://reactnative.dev/docs/virtualizedlist>
- <https://stackoverflow.com/questions/71285002/react-native-flatlist-handling-large-data>
- <https://stackoverflow.com/questions/44384773/react-native-100-items-flatlist-very-slow-performance>