表情键盘优化实践

onscroll_new.html

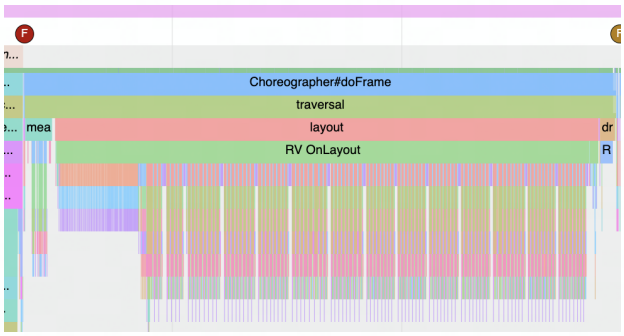
onscroll_original.html

vp_new.html

vp_original.html

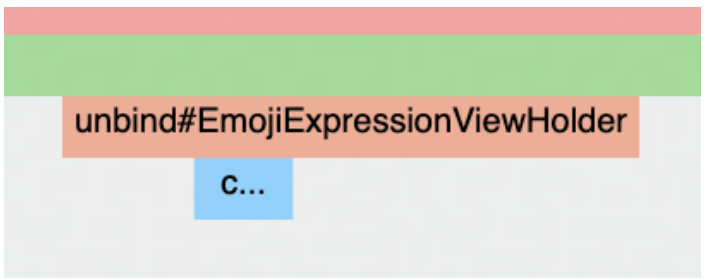
一、分析问题

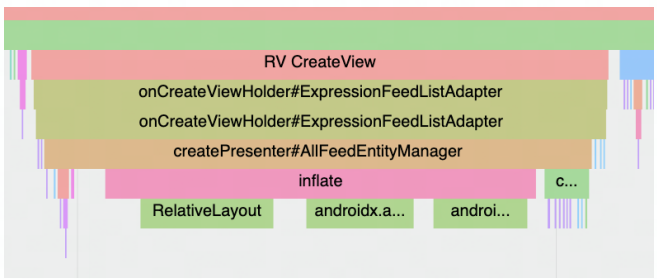
表情页横向切换的viewpager卡顿（从emoji到收藏表情、从收藏表情到网络表情）：

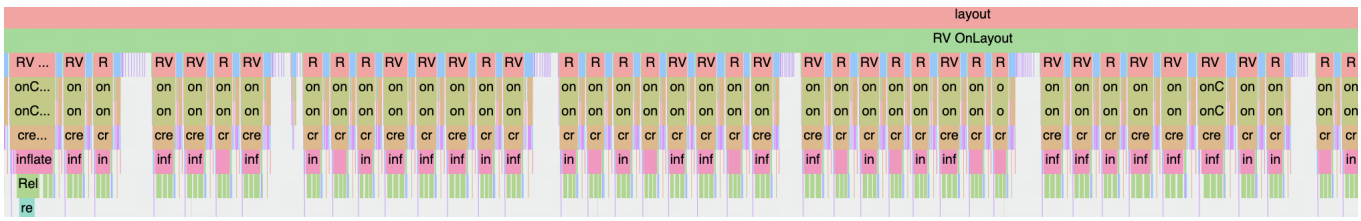


Title	Choreographer#doFrame
Category	android
User Friendly Category	other
Start	10,871.587 ms
Wall Duration	515.503 ms
CPU Duration	512.770 ms
Self Time	0.069 ms
CPU Self Time	0.069 ms

发现其中进行了大量的reyclerview unbind 和 create。

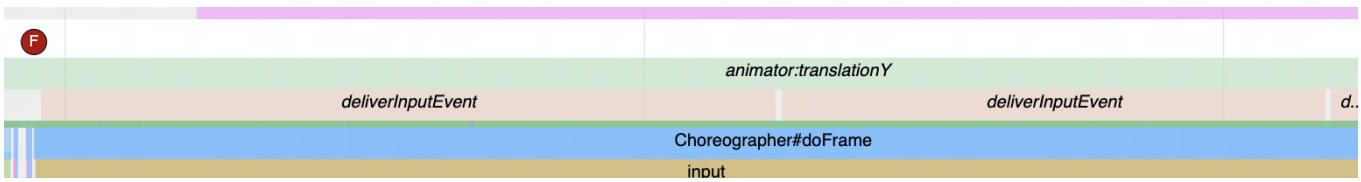


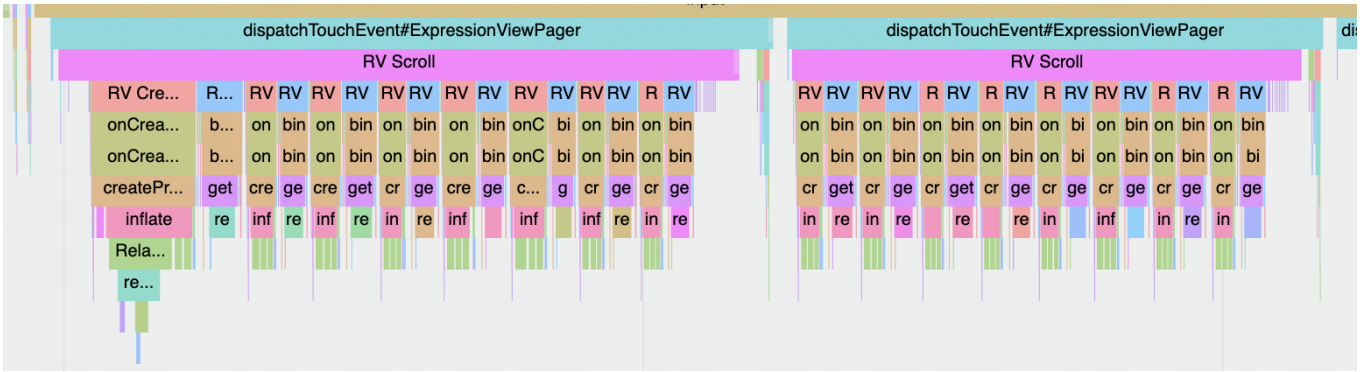




推测是进行了不合理的notifyDataSetChanged()全局更新emoji导致。

emoji表情页下滑也出现了大量的新建view：





推测是没有合理缓存view和emoji的webp图片。

二、处理方式

1. 刷新是为了更新最近使用的emoji，所以viewpager切换时可以只有当进入emoji页才刷新（pagePostion == 0时）。

```
pagerAdapter.addOnSectionChangedListener { sectionPosition, pagePosition, section ->
    // 改变 tab 时要及时刷新 recent 列表
    if (pagePosition == 0) {
        EventBus.post(EmojiExpressionEvent.RefreshRecent())
    }
```

2. 刷新使用DiffUtil，只有有变化时才修改view，可以做到每次增量更新。

```
private fun refreshRecentEmoji() {
    val oldList = ArrayList(expressionModel.expressionItems)
    bindAdapter?.let { it: ExpressionFeedListAdapter
        ExpressionDataManager.refreshRecentEmoji()
        val diffResult = DiffUtil.calculateDiff(EmojiDiffUtil(it.getData(), oldList), detectMoves: true)
        diffResult.dispatchUpdatesTo(it)
    }
```

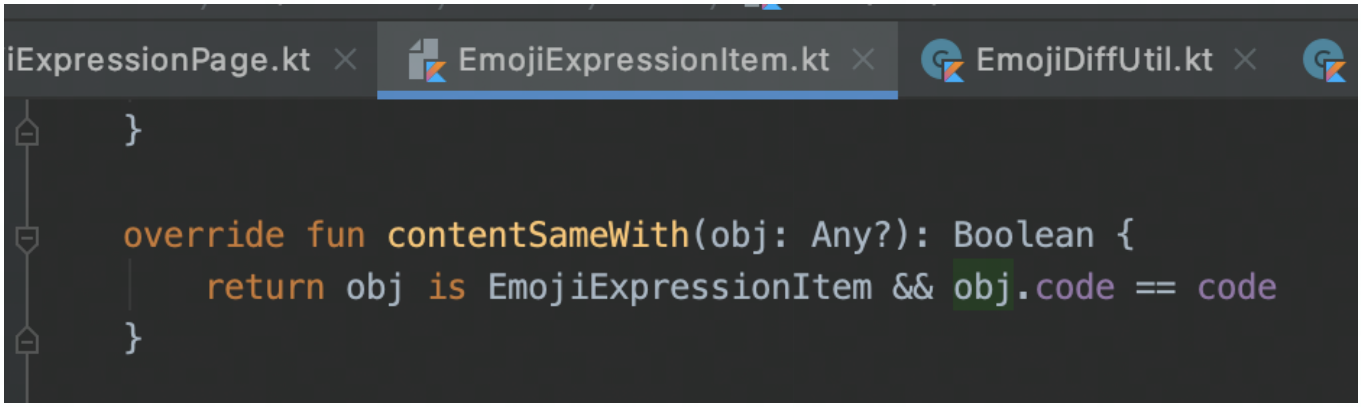
```
class EmojiDiffUtil(val newList: List<AllFeedBaseViewItem>, val oldList: List<AllFeedBaseViewItem>) : DiffUtil.Callback() {

    override fun areItemsTheSame(oldItemPosition: Int, newItemPosition: Int): Boolean {
        return oldList[oldItemPosition].contentSameWith(newList[newItemPosition]) == true
    }

    override fun getOldListSize(): Int {
        return oldList.size
    }

    override fun getNewListSize(): Int {
        return newList.size
    }

    override fun areContentsTheSame(oldItemPosition: Int, newItemPosition: Int): Boolean {
        return oldList[oldItemPosition].contentSameWith(newList[newItemPosition]) == true
    }
}
```



3. 合理使用getExtraLayoutSpace，extraLayoutSpace是layoutManager的参数，它可以延展recyclerview的范围，在不可见的地方事先完成view绘制。不过其绘制方向和scroll方向相同，如果用户频繁上下滑动会起到相反效果。这边暂定一个键盘的高（差不多正好四行emoji）。

```
// 过大的话连续上下滑动会卡
val extraLayoutSpace = KeyBoardUtils.getKeyboardHeight(context)

return object : GridLayoutManager(context, section.perRowExpressionCount) {
    override fun getExtraLayoutSpace(state: State): Int {
        return if (state.hasTargetScrollPosition()) super.getExtraLayoutSpace(state)
        else extraLayoutSpace
        //return super.getExtraLayoutSpace(state)
    }
```

4. 增大ItemViewCacheSize：

```
emojiList.setItemViewCacheSize(expressionModel.section.perRowExpressionCount * 4 + 1)
```

给即将消失的4行emoji一个复活的机会。

5. 增大*recycledViewPool*：

```
val pool = emojiList.recycledViewPool
pool.setMaxRecycledViews(AllFeedEntityManager.getItemViewType(expressionModel.expressionItems[1]), expressionModel.section.perRowExpressionCount * 4)
```

减少onCreateViewHolder调用。

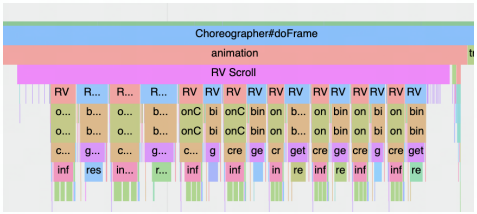
6. 为每张emoji图片做缓存：

```
prefetchEmojiDisposable = Schedulers.io().scheduleDirect {
    emojiExpressionItems.forEach { it: MutableList<EmojiExpressionItem>
        it.forEach { it: EmojiExpressionItem
            EmojiCache.emojiCacheMap[it.drawableId] = ResourceUtil.getDrawable(it.drawableId)
        }
    }
```

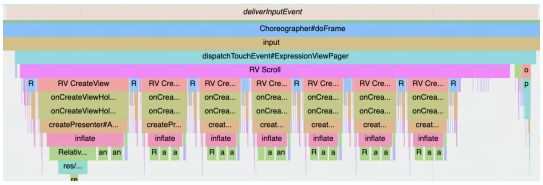
EmojiCache作为单例可以一直在内存，下次bind减少耗时。

三、结论

1. viewpager切换只有在回到emoji keyboard才会刷新，刷新时间从300ms降到24ms。
2. scroll每次bind时间从2ms下降到0.5ms，bind操作每帧做8～16次不等：



Wall Duration	63.437 ms
CPU Duration	62.350 ms
Self Time	0.187 ms
CPU Self Time	0.187 ms



Wall Duration	34.380 ms
CPU Duration	33.267 ms
Self Time	0.083 ms
CPU Self Time	0.083 ms

3. 优化后createview调用大大减少，减少平均时间有待统计。
4. 内存劣化

```
** MEMINFO in pid 15394 [com.zebra.letschat] **
                   Pss  Private  Private  SwapPss      Heap     Heap     Heap
                 Total    Dirty    Clean    Dirty      Size    Alloc     Free
                ------   ------   ------   ------   ------   ------   ------
  Native Heap    77277    77152        0       29    90112    83556     6555
  Dalvik Heap    12478    12452        0       35    23435    11147    12288
Dalvik Other    10404    10404        0        0
       Stack      100      100        0        0
      Ashmem        2        0        0        0
   Other dev      223        0      220        0
    .so mmap    13572     1444     8744        6
    .jar mmap       1        0        0        0
    .apk mmap   27343    10528    15160        0
    .ttf mmap     221        0      180        0
    .dex mmap   41775    37148      828        0
    .oat mmap      52        0        4        0
    .art mmap    6354     5640      120        0
   Other mmap     586        4      392        0
   GL mtrack   17006    17006        0        0
     Unknown    3883     3868        0        0
       TOTAL  211347   175746    25648       70   113547    94703    18843

App Summary
                       Pss(KB)
                       ------
          Java Heap:    18212
        Native Heap:    77152
               Code:    74036
              Stack:      100
           Graphics:    17006
      Private Other:    14888
             System:     9953

               TOTAL:   211347       TOTAL SWAP PSS:       70
```

```
  Native Heap    54588    54532        0    10921    78336    71238     7097
  Dalvik Heap    12659    12640        0      100    22972    10684    12288
Dalvik Other    13016    13016        0       48
       Stack      100      100        0        0
      Ashmem        2        0        0        0
   Other dev      287        0      284        0
    .so mmap    13108      484     9252      936
    .jar mmap       1        0        0        0
    .apk mmap    9553     6044     2016     3348
    .ttf mmap     294        0      284        0
    .dex mmap   23093    18176     1004    14516
    .oat mmap      51        0        4        0
    .art mmap    5574     4828      128      828
   Other mmap     752        4      488        0
   GL mtrack   16939    16939        0        0
     Unknown    2706     2696        0      608
       TOTAL  184028   129459    13460    31305   101308    81922    19385

App Summary
                       Pss(KB)
                       ------
          Java Heap:    17596
        Native Heap:    54532
               Code:    37264
              Stack:      100
           Graphics:    16939
      Private Other:    16488
             System:    41109

               TOTAL:   184028       TOTAL SWAP PSS:    31305
```