

爱吃巧克力的狗

博客园 首页 新随笔 联系 订阅 管理

随笔 - 15 文章 - 4 评论 - 0

React 性能调优记录（上篇），谷歌performance使用

最近工作用的是react，2个项目做下来算是入门啦，哈哈 O(∩\_∩)O，利用空闲时间精读官方文档在性能一节有很多感悟，想写点东西下来，如果能够帮助到正在看的你，我会很开心

第一篇讲一下，谷歌浏览器的performance怎么使用，不感兴趣直接看结论的请看下篇<https://www.cnblogs.com/wzcsqaws/p/10870741.html>

刚开始使用performance的时候，感觉特别复杂，不知道点哪里，网上的教程又太复杂了，就想写个简单的，能用就行，没什么厉害的操作

调试之前首先把代码写好，随便写点复杂的，耗内存的操作，是这么写的（其实是网上抄的 ^\_^)

```
import React,{Component} from 'react'

export class Par extends Component {
  constructor(props) {
    super(props)
    this.state={
      name:'',
      age:'',
      person:[],
    }
  }
  onChange = (stateName, stateValue) => {
    this.setState({[stateName]: stateValue});
  }
  composeChange = (name) => {
    let tt = (e) => {
      this.onChange(name, e.target.value)
    }
    return tt;
  }
  submit(e) {
    const {name,age} = this.state
    var arr = [...this.state.person]
    arr.push({name,age})
    this.setState({
      person:arr
    })
  }

  render() {
    console.log(this.state.person)
    const {name,age,mm} = this.state
    return(
      <div>
        <span>name:</span><input value={name} name='name' type="text"
        onChange={this.composeChange('name')}/>
        <span>age:</span><input value={age} name='age' type="text" onChange=
        {this.composeChange('age')}/>
        <button onClick={this.submit.bind(this)}>确认</button>
        <div>
          { this.state.person.map((e,index)=>(
            <Child key={index} name={e.name} age={e.age}></Child>
          ))
          }
        </div>
      </div>
    )
  }
}

class Child extends Component {
```

公告

昵称: 爱吃巧克力的狗  
园龄: 1年3个月  
粉丝: 1  
关注: 0  
+加关注

< 2019年9月 >						
日	一	二	三	四	五	六
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5
6	7	8	9	10	11	12

搜索

找找看

谷歌搜索

常用链接

- 我的随笔
- 我的评论
- 我的参与
- 最新评论
- 我的标签

随笔档案

- 2019年8月(3)
- 2019年7月(1)
- 2019年6月(2)
- 2019年5月(4)
- 2019年4月(2)
- 2019年3月(2)
- 2018年12月(1)

阅读排行榜

- 1. vue的\$nextTick使用总结, thi s.\$refs为undefined的解决办法, el ement Ui的树形控件setCheckedK eys方法无法使用(1934)
- 2. vue@cli3 项目模板怎么使用pub lic目录下的静态文件,找了好久都不 对, 郁闷! (246)
- 3. Object.assign的用法(224)
- 4. webpack4 插件ProvidePlugin 使用遇到的问题(157)
- 5. webpack4学习之 babel(128)

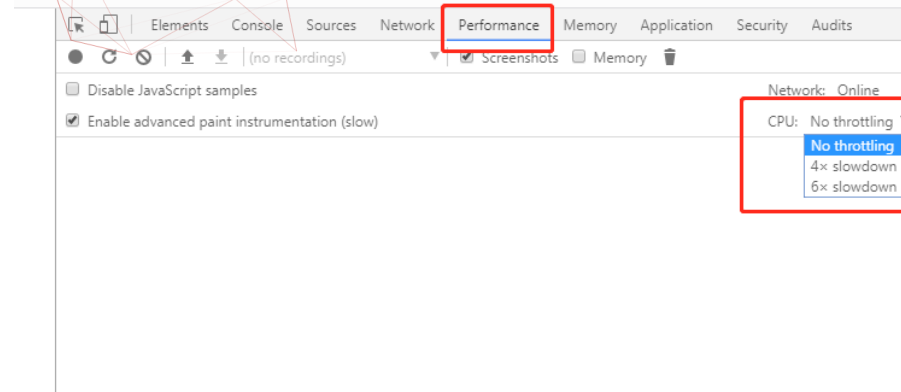
```
componentWillReceiveProps(newProps) {
  console.log(`我新的props的名称是${newProps.name}, age是${newProps.age}。我以前的props的名称是${this.props.name}, age是${this.props.age}是我要re-render了`);
}
render() {
  return (
    <div>
      <span> 姓名: </span>
      <span>{this.props.name}</span>
      <span> 年龄: </span>
      <span>{this.props.age}</span>
    </div>
  )
}
```

copy下来直接运行，这么简单的代码，相信大家都看的懂，就是利用props改变，所有子组件跟着改变来进行损耗性能的操作



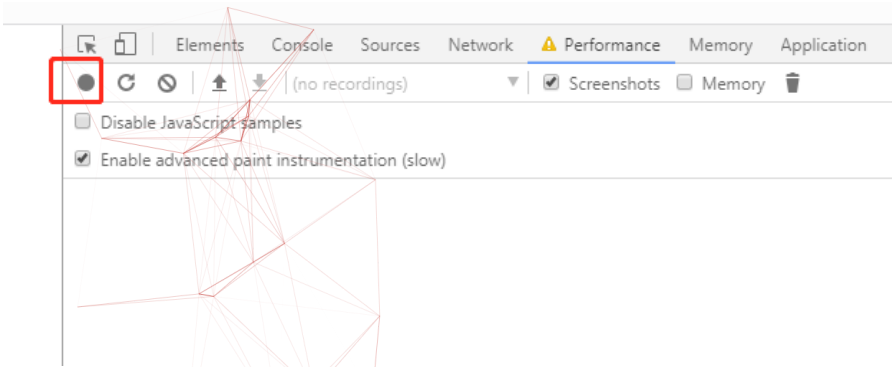
看图，随便写点东西，然后一直点确认，因为子组件越来越多也就渲染越来越慢，这时就可以用上performance了

第一步



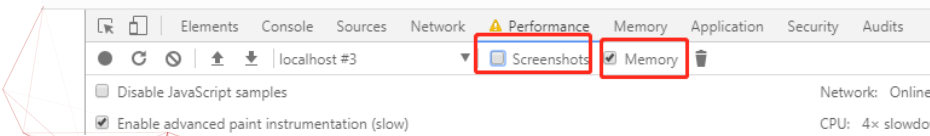
选择performance 然后cpu选择4x slowdown（就是控制cpu的运行速度好让你更方便的测试性能）

第二步

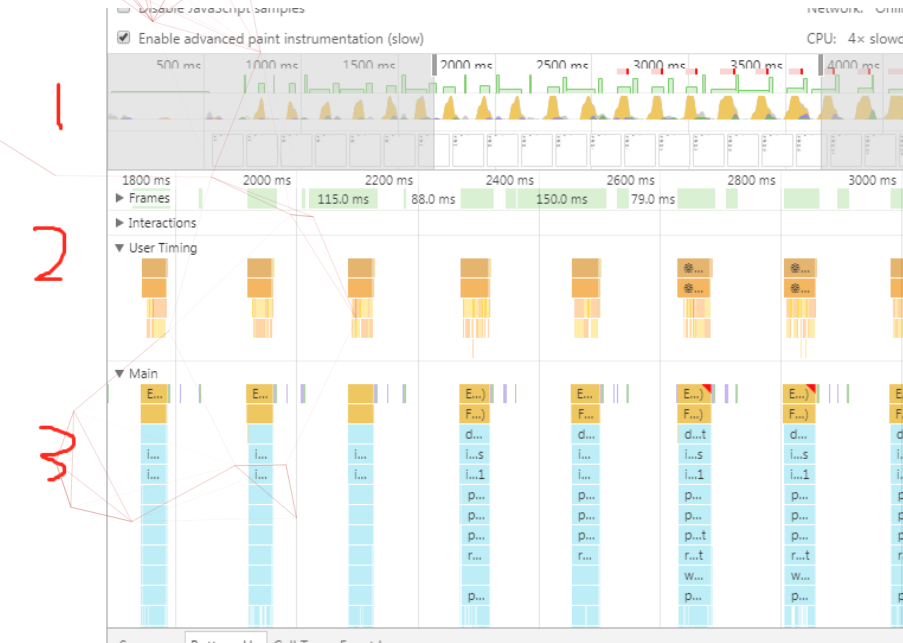


点击这个灰色按钮就可以跑了，然后你就赶紧进行你的耗时操作吧（我这里就是疯狂点确认），点6、7秒就可以点中间的stops了，这步我就不截图了

第三步，终于开始查看了



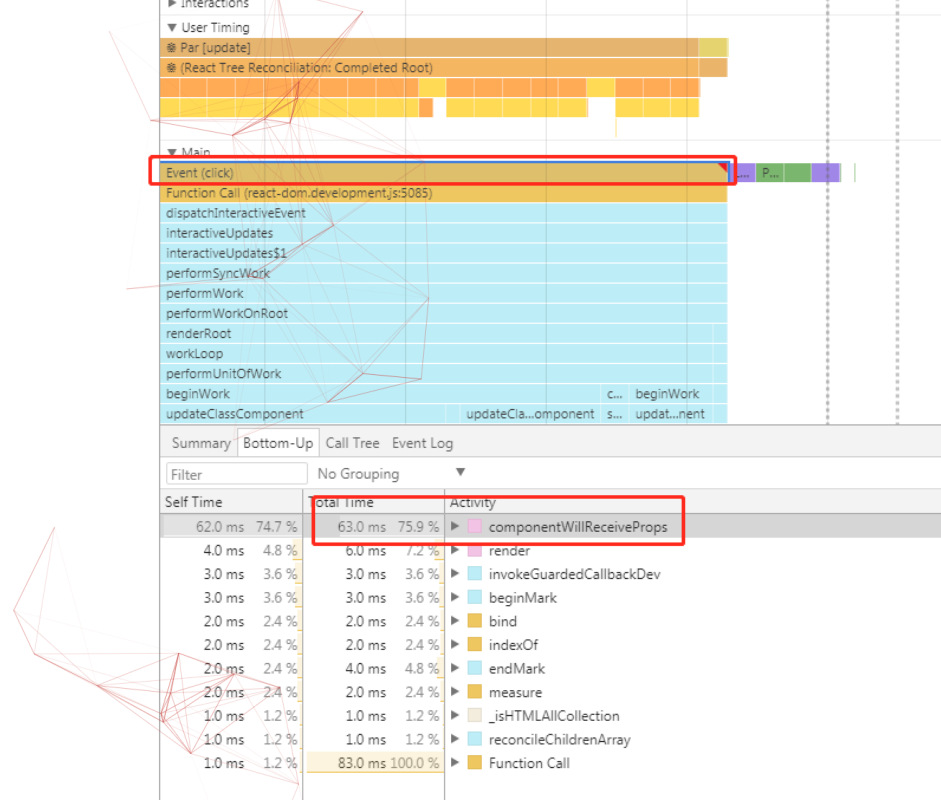
Screenshots是选择是否显示当前页面的缩小版，Memory是查看你的当前内存使用情况



图片上标注的 1 2 3 分别讲解

- 第一部分  
有个时间条，告诉你什么时候你在进行什么操作（大家都知道，我他妈是在说废话吧），黄色的代表js操作，越大自然就表示执行的时候吃性能  
红色的线条表示警告，已经超负荷了，红色越深，性能越慢，可能会出现卡顿等现象，如果观察仔细可以发现一开始是没有红线的，因为子组件少，渲染的次数也少些，自然不是那么耗性能
- 第二部分  
用户的耗时操作会记录在里面，点开看看，你会发现里面有几个你熟悉的react方法

• 第三部分 你进行的操作



点击有右上角有红色的三角的一行，下面就会跟着变化，可以看出是这个  
componentWillReceiveProps生命周期特别耗性能，我们还可以在最右边点击文件打开看

一下到底是哪里特别慢

The screenshot shows the Chrome DevTools Sources tab with the file 'n3.jsx' open. The code includes a class component with state management, a submit function, and a render method. The performance timeline on the left shows various function calls with their durations. A red box highlights the 'componentWillReceiveProps' method call, which has a duration of 629.2 ms, indicating a significant performance bottleneck. Other calls like 'render' and 'setState' are also highlighted with yellow boxes.

## 结论

可以发现componentWillReceiveProps这个生命使用了之后特别特别特别消耗性能，当你把这个生命周期注释后，再多次执行，红色的警告会在6、7倍的子组件数量的时候才会出现所以以后在代码中尽量不要写componentWillReceiveProps哦。这只是一个性能的发现，更多的性能调优的发现请看第二篇

[好文要顶](#)[关注我](#)[收藏该文](#)

爱吃巧克力的狗

关注 - 0

粉丝 - 1

+加关注

« 上一篇: [vue的\\$nextTick使用总结, this.\\$refs为undefined的解决办法, element Ui的树形控件setCheckedKeys方法无法使用](#)

» 下一篇: [React 性能调优记录（下篇），如何写高性能的代码](#)

posted @ 2019-05-15 15:38 爱吃巧克力的狗 阅读(81) 评论(0) 编辑 收藏

[刷新评论](#) [刷新页面](#) [返回顶部](#)

**注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，访问 [网站首页](#)。**

【推荐】超50万C++/C#源码: 大型实时仿真组态图形源码

【活动】[阿里云910会员节多款云产品满减活动火热进行中](#)

【推荐】新手上天翼云，数十款云产品、新一代主机0元体验

【推荐】零基础轻松玩转华为云产品，获壕礼加返百元大礼

【推荐】华为IoT平台开发者套餐9.9元起，购买即送免费课程

#### 相关博文:

- [React性能调优记录（下篇），如何写高性能的代码](#)
- [React中key的作用](#)
- [react 实用的性能优化方式](#)
- [react父子组件通信](#)
- [react.js父子组件通信](#)

#### 最新 IT 新闻:

- 媒体：三星正在生产“更小更便宜”的折叠屏手机
- Cell：带你了解病毒如何感染人类！人类-病毒蛋白质相互作用图谱面世
- 盲人也能网上冲浪：世界上首款盲人笔记本电脑诞生
- 科学家首次对14300年前的“木乃伊狗”进行RNA测序
- 爱吃鸡蛋，到底是好事还是坏事？
- » [更多新闻...](#)

#### 历史上的今天:

2019-05-15 [React 性能调优记录（上篇），谷歌performance使用](#)