

目录

第1章 优化的概念

- 1 开篇词：你的前端性能还能再抢救一下
- 2 解读雅虎35条军规（上）
- 3 解读雅虎35条军规（下）
- 4 你要不要看这些优化指标？

第2章 性能工具介绍

- 5 性能优化百宝箱（上）
- 6 性能优化百宝箱（下）

第3章 网络部分

- 7 聊聊 DNS Prefetch
- 8 Webpack 性能优化两三事
- 9 图片加载优化（上）
- 10 图片加载优化（下）

第4章 缓存部分

- 11 十八般缓存
- 12 CDN 缓存
- 13 本地缓存（Web Storage）
- 14 浏览器缓存（上）
- 15 浏览器缓存（下）

第5章 渲染部分

- 16 渲染原理与性能优化
- 17 如何应对首屏“一片空白”（上）
- 18 如何应对首屏“一片空白”（下）

26 不可不知的React、Vue 优化技巧

更新时间：2019-09-10 09:52:07



“

学习知识要善于思考，思考，再思考。

—— 爱因斯坦

”

引入

相信现在大部分前端业务开发使用的框架都离不开React、Vue，很多同学对自己的要求都是能用这个框架即可，而忽略了如何写性能更高效，那么这一节主要介绍平时开发当中经常会用到的React、Vue优化技巧。

React

简介

React原本是Facebook内部开发的前端框架，并于2013年开源。React的出现将前端开发带到了一个全新的高度，说React是前端中最受关注的前端框架一点也不为过，React当中的很多思想都对后来的框架(Vue)有着深远的影响。但是很多人对React的掌握仅仅停留在可以使用的阶段，对于其中的一些优化方法并不能灵活应用，那么下面我们就来介绍关于React优化技巧。

如何优化

- 避免重复渲染，大家都知道只要状态发生变化(state，props)，那么组件就会重新渲染，对于重复渲染我们有可以通过下面一些方法来避免：
 1. 使用shouldComponentUpdate生命周期钩子函数来对比前后的state和props，看是否变化，进而确定是否需要重新渲染。如果你的组件非常简单，状态只由props来决定，我们也可以使用PureComponent，来满足大部分的场景。如下：

```
class Test extends React.PureComponent {  
  render() {  

```

目录	
第1章 优化的概念	
1 开篇词：你的前端性能还能再抢救一下	
2 解读雅虎35条军规（上）	
3 解读雅虎35条军规（下）	
4 你要不要看这些优化指标？	
第2章 性能工具介绍	
5 性能优化百宝箱（上）	
6 性能优化百宝箱（下）	
第3章 网络部分	
7 聊聊 DNS Prefetch	
8 Webpack 性能优化两三事	
9 图片加载优化（上）	
10 图片加载优化（下）	
第4章 缓存部分	
11 十八般缓存	
12 CDN 缓存	
13 本地缓存（Web Storage）	
14 浏览器缓存（上）	
15 浏览器缓存（下）	
第5章 渲染部分	
16 渲染原理与性能优化	
17 如何应对首屏“一片空白”（上）	
18 如何应对首屏“一片空白”（下）	
19 不容小觑的 DOM 性能优化	

```
    </div>
  )
}
}
```

- 2.尽量使用函数式组件，这样可以尽量避免没有必要的渲染。
- 不要在渲染函数当中放置无用代码，我们都知道Render函数就是一个负责渲染的纯函数，所以一些无用的逻辑代码一定放在render函数外面。
 - 在遍历列表的时候添加key属性，相信这一点大家都有体会，如果我们没有添加key属性，那么控制台会报warning。key主要作用就是为了实现组件的复用，最好不要使用数组的索引，而要使用其唯一的身份标识，因为如果是中间元素发生变化，由于索引的特殊性，后面所有元素都需要进行更新，因此尽量使用它的唯一身份标识。
 - 使用stateless component，也就是我们常说的函数式组件，它没有生命周期方法，只有一个render方法，对于一些简单组件完全可以使用函数式组件来完成，React内部对于这类组件有专门的优化。
 - 谨慎使用setState，开发当中如果遇到和视图无关的，但有变化的数据，不要放在state里面，这个时候我们可以直接写成组件内私有变量或者组件实例上的变量。如下：

```
class Test extends Component {
  record = false
  onMouseDown = () => {
    this.record = true;
  }
  onMouseUp = () => {
    this.record = false;
  }
  render() {
    return (<div>
      <button onMouseDown={this.onMouseDown} onMouseUp={this.onMouseUp} />
      <div/>);
  }
}
```

- 使用不可变数据，使用不可变数据让状态变得可预测，可选用immutable.js，它是Facebook官方推出的持久性数据结构的库，保证了结构共享和持久性数据结构，对于提高性能有着非常大的帮助。具体的使用方法可以去immutable.js[官方网站](#)进行详细学习。
- 简化props，复杂的props会使组件难以维护，而且如果props的数据结构过于复杂，也会影响shallowCompare，导致组件无法准确刷新，所以当你的props过于复杂的时候，就要考虑将组件进行拆分。

Vue

简介

Vue是渐进式的JavaScript框架，和React一样是现在前端开发当中应用最广泛的框架，并且一样都是只关注视图层的框架，非常容易上手，而且方便与第三库进行整合。另外，当与现代化的工具链以及各种支持类库结合使用时，Vue也完全能够为复杂的单页应用提供驱动。下面我们来介绍与Vue有关的性能优化方法。

← 慕课专栏			☰ 你不知道的前端性能优化技巧 / 26 不可不知的React、Vue 优化技巧		
目录					
第1章 优化的概念					
1 开篇词：你的前端性能还能再抢救一下			• 首先我们介绍一个和React一样的优化方法，同样也是遍历列表的时候添加key属性，原因和React一样，底层都是通过虚拟DOM比较前后变动，然后应用到真正的DOM当中。		
2 解读雅虎35条军规（上）					
3 解读雅虎35条军规（下）					
4 你要不要看这些优化指标？			• 使用异步组件，在大型应用当中，我们常常需要使用代码分割这种方式，来避免过长时间的加载，而Vue官方正好为我们提供了异步组件这种方式，来保证只有在我们需要的时候才从服务端拉取。方法如下：		
第2章 性能工具介绍			<pre>Vue.component('async-example', function (resolve, reject) { setTimeout(function () { // 向 `resolve` 回调传递组件定义 resolve({ template: '<div>I am async!</div>' }) }, 1000) })</pre>		
5 性能优化百宝箱（上）			Vue异步组件同样也是设计模式当中的工厂模式的经典运用，这里大家要多注意，它是面试当中的高频考题。		
6 性能优化百宝箱（下）			• 提取组件当中的CSS到单独文件，如果把JavaScript代码和CSS代码放到一起，运行时CSS会以style标签的方式通过JavaScript注入，这对运行时的性能有一定影响，因此CSS最好单独分离出来。		
第3章 网络部分			• 使用Function-based API，其实在今年的VueConf大会上，Vue作者尤雨溪就宣布之后Vue会撤销Class API，逐步向Function-based API过渡，原因是相比Class API，Function-based API有更好的TypeScript 类型推导支持，更灵活的逻辑复用能力，Tree-shaking 友好，代码更容易被压缩等一系列优势，其实这与React大力推行的hooks是极其类似的，关于具体使用方法，可以参考 这里 。		
7 聊聊 DNS Prefetch					
8 Webpack 性能优化两三事			• 使用扁平化的store数据结构，很多时候我们接口返回的数据结构嵌套非常深，如果我们把这样的数据直接存储到store当中，我们取的时候非常不方便，对性能也会造成一定影响，所以我们将数据结构处理成扁平化结构再进行存储。		
9 图片加载优化（上）					
10 图片加载优化（下）					
第4章 缓存部分					
11 十八般缓存					
12 CDN 缓存					
13 本地缓存（Web Storage）					
14 浏览器缓存（上）					
15 浏览器缓存（下）					
第5章 渲染部分					
16 渲染原理与性能优化			小结		
17 如何应对首屏“一片空白”（上）			这一节我们介绍了Vue和React的一些性能优化方法，因为是前端使用最广泛的两大框架，它们的优化方法大家下去要多进行相关实践。虽然说框架提高了生产力，但是如果使用不当可能会适得其反。因此，在日常的开发当中，大家不应该只满足于会用框架，更要追寻框架背后的一些东西。		
18 如何应对首屏“一片空白”（下）			← 25 移动端的优化技巧也想让你知道 27 写在最后 →		
19 不容小觑的 DOM 性能优化			精选留言 0		
			欢迎在这里发表留言，作者筛选后可公开显示		

← 慕课专栏	☰ 你不知道的前端性能优化技巧 / 26 不可不知的React、Vue 优化技巧
目录	目前暂无任何讨论
第1章 优化的概念	
1 开篇词：你的前端性能还能再抢救一下	千学不如一看，千看不如一练
2 解读雅虎35条军规（上）	
3 解读雅虎35条军规（下）	
4 你要不要看这些优化指标？	
第2章 性能工具介绍	
5 性能优化百宝箱（上）	
6 性能优化百宝箱（下）	
第3章 网络部分	
7 聊聊 DNS Prefetch	
8 Webpack 性能优化两三事	
9 图片加载优化（上）	
10 图片加载优化（下）	
第4章 缓存部分	
11 十八般缓存	
12 CDN 缓存	
13 本地缓存（Web Storage）	
14 浏览器缓存（上）	
15 浏览器缓存（下）	
第5章 渲染部分	
16 渲染原理与性能优化	
17 如何应对首屏“一片空白”（上）	
18 如何应对首屏“一片空白”（下）	
19 不容小觑的 DOM 性能优化	