

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



2017

Tree Shaking 性能优化实践

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



林溪



林溪

9年web 开发者，2013年加入百度

百度资深前端工程师

百度外卖前端架构师





目录

- 什么是tree shaking
- Tree shaking原理
- Tree shaking实践
- 总结





什么是tree shaking





去掉用不到的多余代码





支持tree shaking的构建工具



Rollup

Webpack2

Closure compiler





Tree Shaking 原理



Dead Code Elimination

Dead Code

- 代码不会被执行，不可到达
- 代码执行的结果不会被用到
- 代码只会影响死变量（只写不读）

无用的模块消除

ES6 module 特点：

- 只能作为模块顶层的语句出现
- import 的模块名只能是字符串常量
- import binding 是 immutable 的

依赖关系是确定的，和运行时的状态无关
可以进行可靠的静态分析，然后进行消除

1

DCE试验

.babelrc

```
.babelrc x
1 {
2   "presets": [
3     ["es2015", { "modules": false }]
4   ]
5 }
```

```
1  let foo = () => {
2    let x = 1
3    if (false) {
4      console.log('never reached')
5    }
6    let a = 3
7    return a
8  }
9
10 let baz = () => {
11   var x = 1
12   console.log(x)
13   function unused () {
14     return 5
15   }
16   return x
17   let c = x + 3
18   return c
19 }
20
21 baz()
```

```
1 let foo = () => {
2   let x = 1
3   if (false) {
4     console.log('never reached')
5   }
6   let a = 3
7   return a
8 }
9
10 let baz = () => {
11   var x = 1
12   console.log(x)
13   function unused () {
14     return 5
15   }
16   return x
17   let c = x + 3
18   return c
19 }
20
21 baz()
```

rollup.bundle.js x

```
1 'use strict';
2
3 var baz = function baz() {
4   var x = 1;
5   console.log(x);
6   return x;
7   var c = x + 3;
8   return c;
9 };
10
11 baz();
```

webpack.bundle.js x

```
63 /******/ return __webpack_require
64 /******/ })
65 /******/ (
66 /******/ ([
67 /* 0 */
68 /******/ (function(module, exports)
69
70 var foo = function foo() {
71   var x = 1;
72   if (false) {
73     console.log('never reached');
74   }
75   var a = 3;
76   return a;
77 };
78
79 var baz = function baz() {
80   var x = 1;
81   console.log(x);
82   function unused() {
83     return 5;
84   }
85   return x;
86   var c = x + 3;
87   return c;
88 };
89
90 baz();
```

125B

3KB

Rollup自帶部分DCE
Webpack 不做DCE

Rollup/Webpack + Uglify

The image shows a comparison of Rollup and Webpack bundling. It consists of three main panels:

- main.js (Source):** Contains two functions: `foo` (with a dead code branch) and `baz` (with a `used` variable and a `unused` function).
- rollup.config.js (Configuration):** Configures Rollup to use Babel and Uglify plugins, outputting to `rollup.bundle.js` in CJS format with tree shaking and Uglify optimization.
- rollup.bundle.js (Output):** Shows the result of Rollup bundling, which is 64B. The code is minified and includes the `baz` function and its call.
- webpack.config.js (Configuration):** Configures Webpack to use Babel and Uglify plugins, outputting to `webpack.bundle.js` in CJS format with Uglify optimization.
- webpack.bundle.js (Output):** Shows the result of Webpack bundling, which is 517B. The code is minified but includes a large amount of dead code from the `foo` function and the `unused` function.

Uglify实现DCE 

2

无用代码消除

- 函数消除试验
- 类消除试验

```
webpack.bundle.js x
post
[0])return t[0].exports, var t={};n.m=e,n.c=t,
[o]={i:o,l:!1,exports:{}};
return e[o].call(r.exports,r,
r.exports,n),r.l=!0,r.exports}
n.d=function(e,t,o){n.o(e,t)||
Object.defineProperty(e,t,
{configurable:!1,enumerable:!0,
get:o}}),n.n=function(e){var
t=e&&e.__esModule?function()
{return e.default}:function()
{return e};return n.d(t,"a",t),
t},n.o=function(e,n){return
Object.prototype.hasOwnProperty.
call(e,n)},n.p="",n(n.s=0)}(
[function(e,n,t){"use strict";
function o(){console.log("do
post")}Object.defineProperty(n,
"__esModule",{value:!0});
!function(){o();console.log(1)}
()]];
```

无用函数消除成功 

```

1 !function(n){function e(o){if(t[o])return t[o]
  .exports;var r=t[o]={i:o,l:!1,exports:{}};return n[o]
  .call(r.exports,r,r.exports,e),r.l=!0,r.exports}var
  t={};e.m=n,e.c=t,e.d=function(n,t,o){e.o(n,t)||
  Object.defineProperty(n,t,{configurable:!1,
  enumerable:!0,get:o})},e.n=function(n){var t=n&&
  n.__esModule?function(){return n.default}:function()
  {return n};return e.d(t,"a",t),t},e.o=function(n,e)
  {return Object.prototype.hasOwnProperty.call(n,e)},
  e.p="",e(e.s=0)}([function(n,e,t){"use strict";
  function o(){console.log("do post")}function r(n,e)
  {if(!(n instanceof e))throw new TypeError("Cannot
  call a class as a function")}Object.defineProperty(e,
  "__esModule",{value:!0});var i=function(){function n
  (n,e){for(var t=0;t<e.length;t++){var o=e[t];
  o.enumerable=o.enumerable||!1,o.configurable=!0,
  "value" in o&&(o.writable=!0),Object.defineProperty(n,
  o.key,o)}}return function(e,t,o){return t&&n
  (e.prototype,t),o&&n(e,o),e}}();!function(){function
  n(){r(this,n),this.display="none"}i(n, [{key:"show",
  value:function(){this.display="block"}}, {key:"hide",
  value:function(){this.display="none"}}, {key:"isShow",
  value:function(){return"block"===this.display}}]);
  !function(){o();console.log(1)}()});

```

消除失败



D3

Rollup {treeshaking: false}

230KB

Rollup {treeshaking: true}

96KB

Webpack3

220KB



```
1
2 import { select } from 'd3'
3
4 var svg = select('svg')
5 svg.style('background', 'black')
6 svg.append('circle')
7   .attr('id', 'circle01')
8   .attr('cx', 200)
9   .attr('cy', 200)
10  .attr('r', 100)
11  .style('fill', 'lightgreen')
12
```


Three.js

Rollup {treeshaking: false}

512KB

Rollup {treeshaking: true}

492KB

Webpack3

474KB

```
7 import {  
8   Vector3,  
9   Scene,  
10  Clock,  
11  PerspectiveCamera,  
12  WebGLRenderer,  
13  PointLight,  
14  MeshBasicMaterial,  
15  MeshFaceMaterial,  
16  DoubleSide,  
17  PlaneGeometry,  
18  Mesh,  
19  CubeGeometry,  
20  FogExp2  
21 } from 'three'
```

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



What Happened?



Victorystick commented on 11 May 2016

Owner



Indeed. Rollup only does tree-shaking for functions and variables that use `import / export` syntax. You're correct in assuming that classes can't have their methods tree-shaken; see [#349](#).



kzc commented on 2 Dec 2016

Contributor



Although the simple program above can be statically analyzed, in a more complex program any object property can be accessed in a dynamic fashion that cannot always be statically deduced:

```
var Person = (function(){
  var Person = function(){};
  Person.prototype.run = function run() { console.log("run"); }
  Person.prototype.jump = function jump() { console.log("jump"); }
  return Person;
})();
var p = new Person();
p.jump();

p[ Math.random() > 0.5 ? "run" : "jump" ]();
```

which is why rollup errs on the side of caution and retains all methods.



- Rollup只处理函数和import / export变量
- Javascript动态语言特性使得分析比较困难
- Side Effect广泛存在



Tree Shaking效果不佳

Tree Shaking对顶层纯函数效果更好



<https://github.com/rollup/d3-jsnext>



Closure Compiler

编译命令

```
java -jar ./libs/closure-compiler-v20170626.jar --  
compilation_level ADVANCED_OPTIMIZATIONS --  
process_common_js_modules --transform_amd_modules --  
-language_in ECMASCRIPT6 --js src/main.js --js  
src/util.js --js src/menu.js --js_output_file  
cc.bundle.js --process_common_js_modules
```

输出结果

JS cc.bundle.js ✕

```
1 console.log("do post");console.log(1);|  
2
```

```
js menu.js  js util.js  x
1 goog.provide('util')
2
3 export function post () {
4   console.log('do post')
5 }
6
7 post.prototype.before = () => {
8   console.log('before')
9 }
10
11 export function get () {
12   console.log('do get')
13 }
14
```

侵入式

Annotating JavaScript for
the Closure Compiler

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



Tree Shaking对web意义重大
但目前还处在发展阶段

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



Tree Shaking 实践

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



Webpack 2升级



Webpack 3升级



问题

- 效果不明显
- 无用的代码并没有完全消除
- rollup也不能根本解决问题
- Closure compiler有侵入，需要改代码

babel-plugin-import-fix

<https://github.com/lin-xi/babel-plugin-import-fix>

减小import范围

<pre>import ReactDOM from 'react-dom' import { Button } from 'antd'; ReactDOM.render(<div> <Button onClick={onClick} /> </div> , document.body); function onClick() { console.log('click'); }</pre>	<pre>import ReactDOM from 'react-dom' import {Button} from 'antd/lib/button'; import 'antd/lib/style'; ReactDOM.render(<div> <Button onClick={onClick} /> </div> , document.body); function onClick() { console.log('click'); }</pre>
---	---

1.5MB

310KB



```
{
  visitor: {
    Program(path, { opts = {} }) {}, //根结点
    ImportDeclaration() {}, //import语句
    CallExpression() {}, //函数调用表达式
    MemberExpression() {}, //对象属性表达式
    VariableDeclarator() {}, //变量定义语句
    LogicalExpression() {}, //逻辑表达式
    ConditionalExpression() {}, //三目运算表达式
    IfStatement() {}, //if语句
    ExpressionStatement() {}, //表达式语句
    ReturnStatement() {}, //return语句
    ExportDefaultDeclaration() {}, //export default 语句
    BinaryExpression() {}, //算术表达式
  }
}
```





支持库



XCUI





Three.js

pixi.js

echarts.js



为了能被shake：

- 使用ES6模块规范 [静态分析]
- 构建模块化的输出 [package.json module字段]
- 组件模块独立输出 [import-fix]

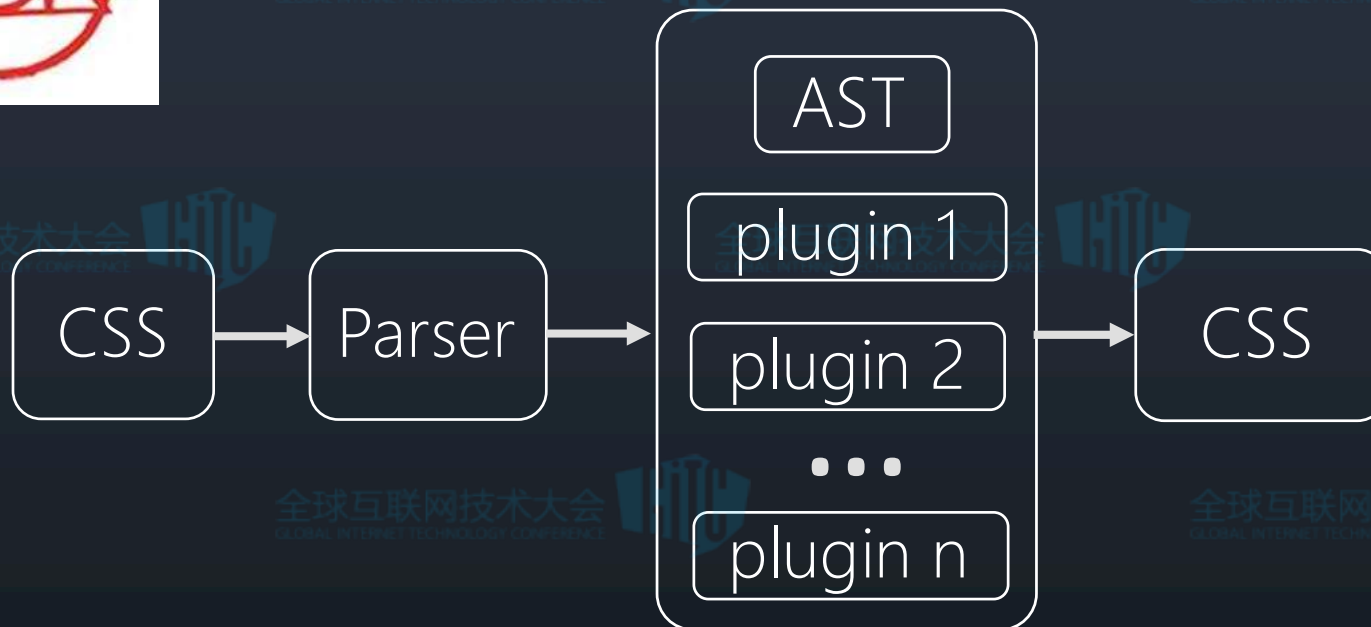


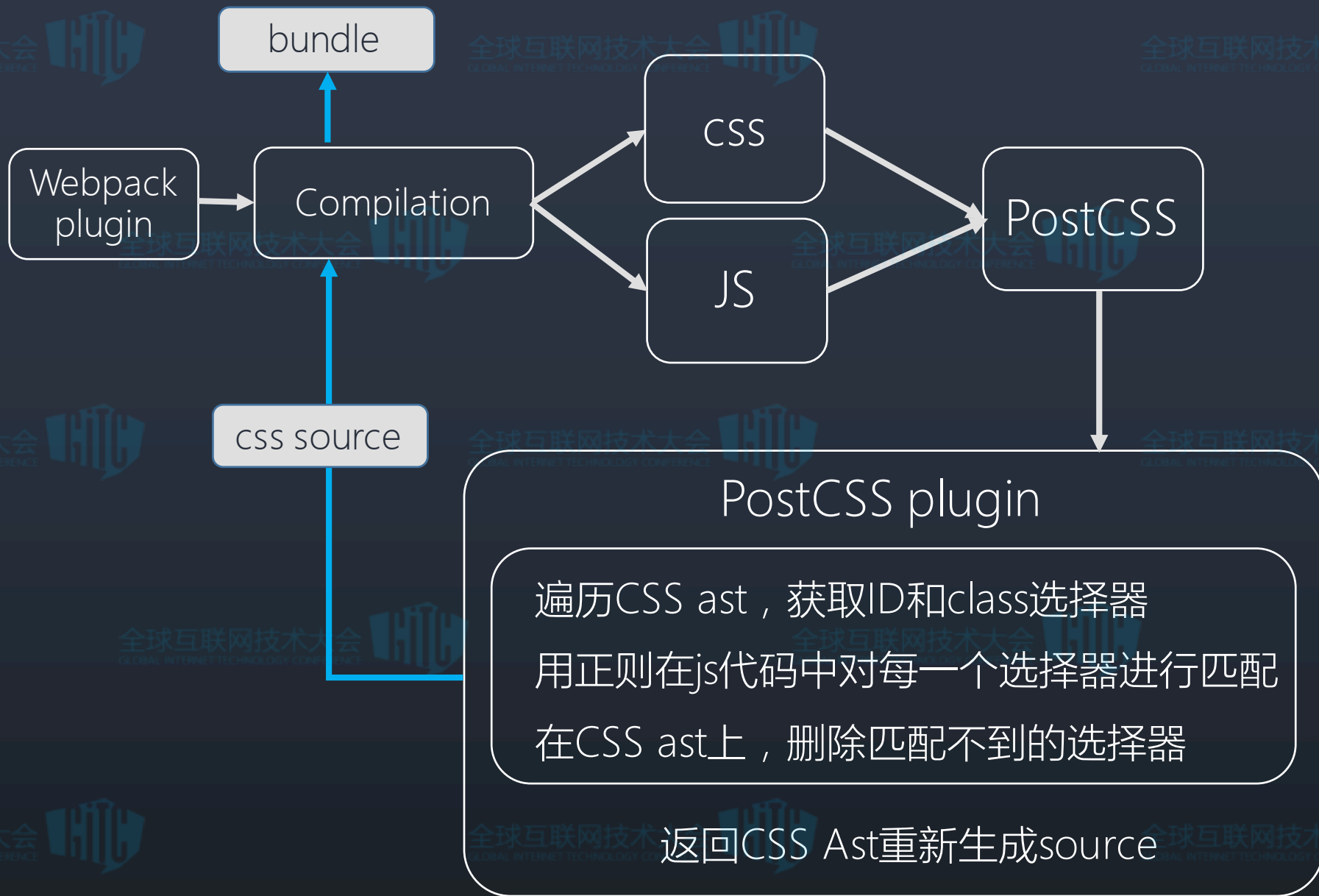
webpack-css-treeshaking-plugin

<https://github.com/lin-xi/webpack-css-treeshaking-plugin>



PostCSS







- Side Effect 也存在

```
1 <template>
2   <div class="compo-paper">
3     <div class="{ 'page-header': showHeader}">
4       <div class="user-name">
5         </div>
6     </div>
7   </div>
8 </template>
```

Vue

```
2 render(){
3   let headerClash = showHeader ? 'headerClass' : '';
4   return (
5     <div class="compo-paper">
6       <div class="{headerClass}">
7         <div class="user-name">
8           </div>
9       </div>
10    </div>
11  )
}
```

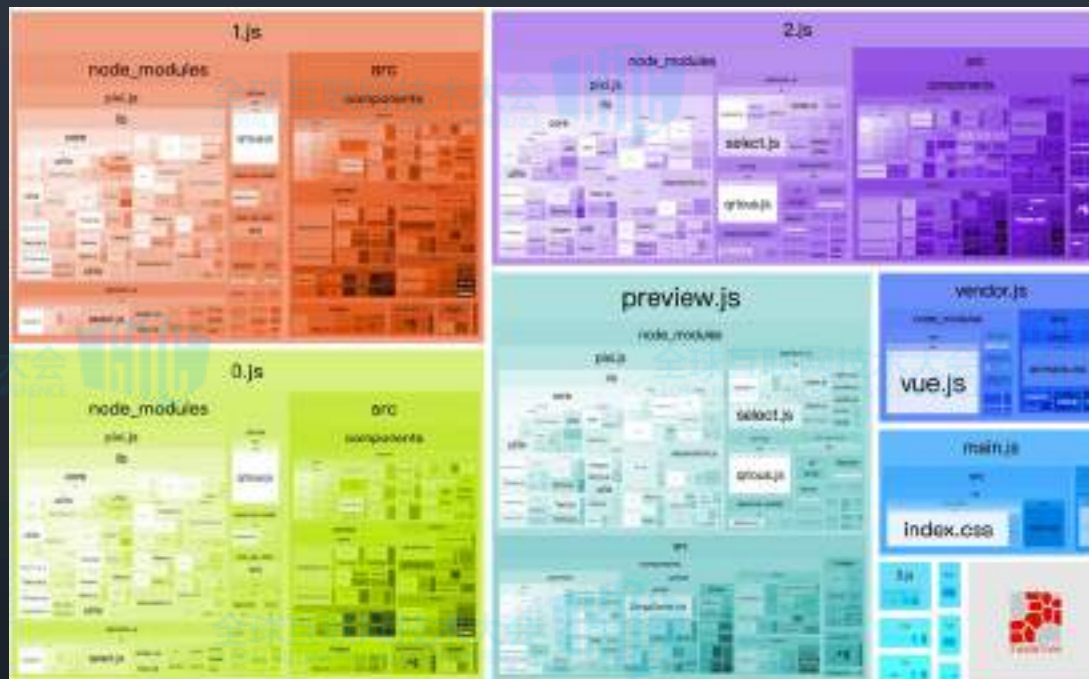
React



广义Tree Shaking, bundle去重

1

webpack-bundle-analyzer



2

CommonsChunkPlugin

```
entry: {  
  main: [path.join(__dirname, './src/main.js')],  
  preview: [path.join(__dirname, './src/preview.js')],  
  vendor: ['vue']  
},
```

```
new webpack.optimize.CommonsChunkPlugin({  
  name: 'vendor',  
}),
```

问题

- Vendor需要不停的手动维护
- 不能使用babel-plugin-import-fix插件

自动维护通用模块

```
new webpack.optimize.CommonsChunkPlugin({
  name: 'vendor',
  minChunks: (module, count) => [
    (module.context && module.context.indexOf('node_modules') !== -1)
    || count >= 2
  ]
}),
```

- 去除vendor entry
- 自动将所有的node_module和出现两次及以上的文件打包
- babel-plugin-import-fix 生效

问题

- 异步模块没有得到优化

```
new webpack.optimize.CommonsChunkPlugin({
  name: 'vendor',
  minChunks: (module, count) => [(
    (module.context && module.context.indexOf('node_modules') !== -1)
    || count >= 2
  )]
}),
new webpack.optimize.CommonsChunkPlugin({
  names: ['main', 'preview', 'EditPage', 'MinePage', 'Mine
  async: 'async-common',
  minChunks: 2
}),
```

Perfect !
彻底去重



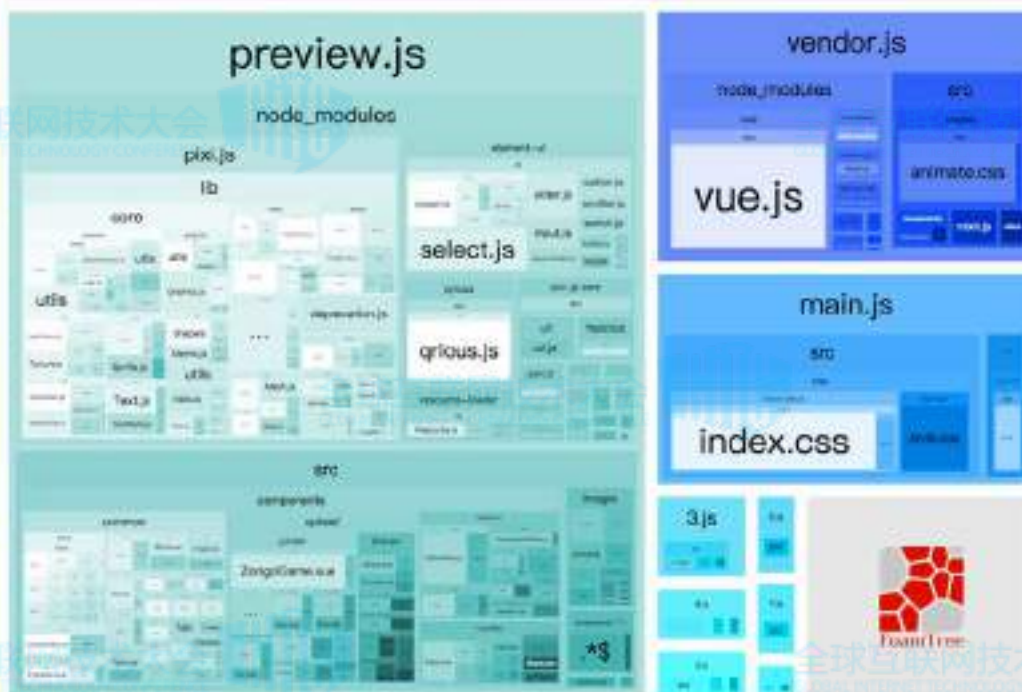
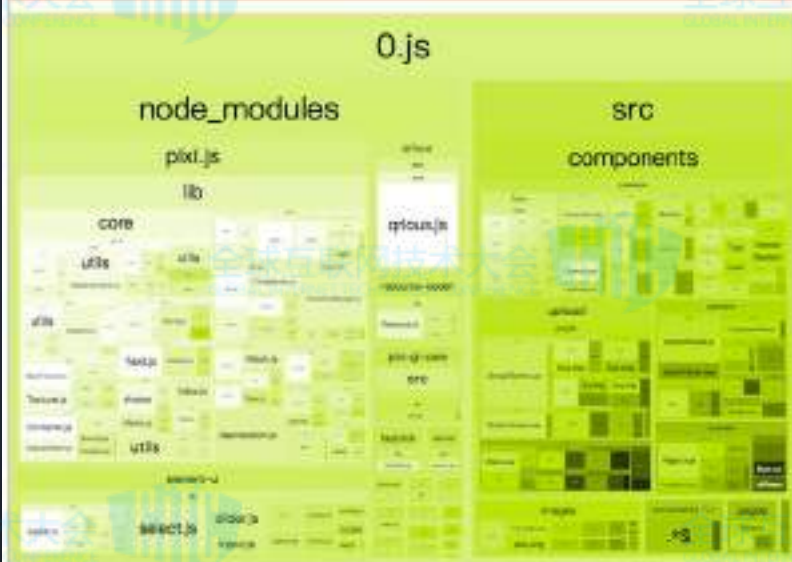
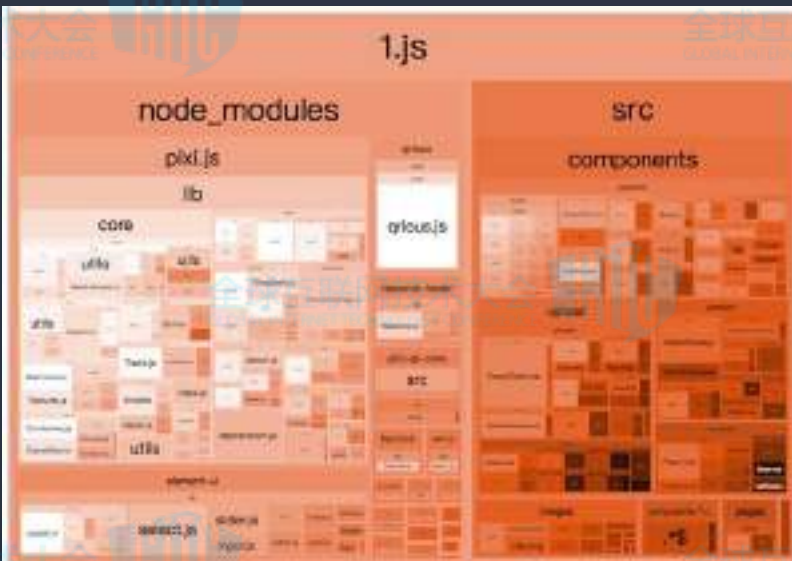
Show chunks:

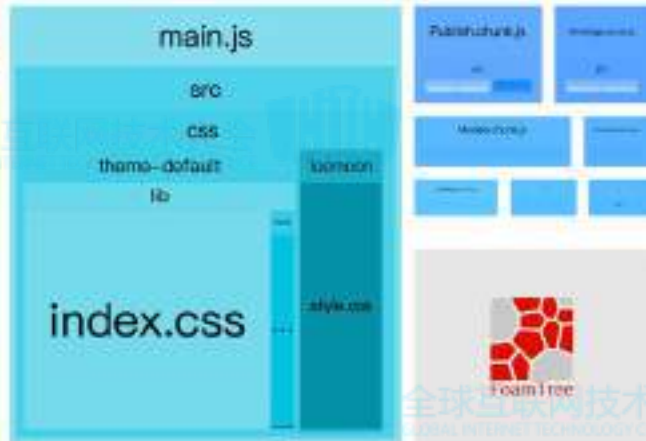
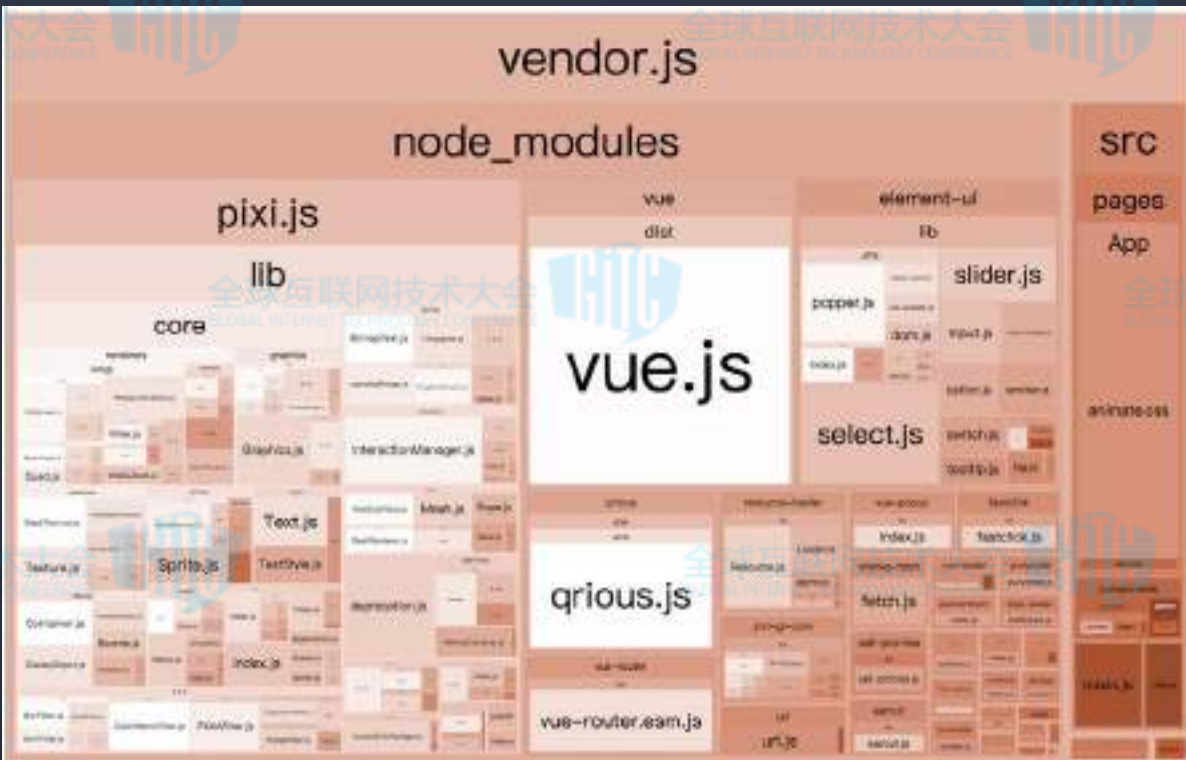
- All (**4.29 MB**)
- preview.js (**1.11 MB**)
- MinePage.chunk.js (**938.51 KB**)
- Models.chunk.js (**937.62 KB**)
- EditPage.chunk.js (**930.4 KB**)
- main.js (**386.77 KB**)
- Publish.chunk.js (**18.99 KB**)
- MinePublish.chunk.js (**12.74 KB**)
- MediaUpload.chunk.js (**11.8 KB**)
- MineWidget.chunk.js (**8.93 KB**)
- MineComponent.chunk.js (**8.2 KB**)
- RealTime.chunk.js (**4.51 KB**)

Show chunks:

- All (**1.71 MB**)
- vendor.js (**779.8 KB**)
- async-common-main.chunk.js (**381.74 KB**)
- preview.js (**380.34 KB**)
- main.js (**154.72 KB**)
- Publish.chunk.js (**13.06 KB**)
- MinePage.chunk.js (**9.97 KB**)
- Models.chunk.js (**9.06 KB**)
- MinePublish.chunk.js (**4.17 KB**)
- EditPage.chunk.js (**3.98 KB**)
- MediaUpload.chunk.js (**3.24 KB**)
- MineWidget.chunk.js (**3 KB**)
- MineComponent.chunk.js (**2.27 KB**)
- RealTime.chunk.js (**2.08 KB**)









总结



Tree shaking 做了些什么

Tree shaking 适用场景

Babel插件减少import范围

Css tree shaking

Bundle去重

使用ES6模块

Tree Shaking不能解决所有问题，重视Eslint

减少side effect，无状态函数式编程



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



谢谢

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



我的知乎

百度外卖专栏



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE

