

有生命力的复合型前端铁军实践

小春 - 摩拜前端负责人

前滴滴公共前端团队负责人 - 负责核心公共平台团队和小巴等业务团队
在之前最近的在豌豆荚经历过基本所有的业务。

喜欢研究和实践：

13年开始应用react和angular到业务中，国内较早开始布道普及vue.js。

喜欢写写：

出版了包含最近的《Vue.js权威指南》，即将面世的css翻译书籍以及新书编写。
在infoq等发稿，iteye写了不到2k篇技术文章

喜欢分享：

担任架构大会前端出品人：GIAC 全球互联网架构大会
参加了多次重大的技术峰会 - 2016 CSDN 举办的 SDCC、全球首场Vue.js分享会vueconf等

大纲：

1、生命力

2 W 1 H 1 O 1 V原则 – What、Why、How、Optimize、Verification

2、复合型

devOps + 多方案的服务化 + 网络通信 + 数据量化意识 + 平台化手段

3、铁军

差异化、育人

范围选择：

Vue

egg alinode

Ssr nuxt

国际化 + 自动化测试

多页应用

QUIC > http2

YML

Docker > docker-compose.yml > Dockerfile > 常用命令

Ansible

Electron

私服

Gitlab CI

Vue.js 业务框架设计和优化

- 1、多页应用(内混 SPA 或纯多页)
- 2、多页应用 + 国际化构建支持
- 3、运营类业务应用
- 4、ssr 多品类线上业务实践
 - Nuxt + express 以及如何支持国际化
 - Egg + webpack2/3 + Vue
 - 服务监控

何为多页？

```
1 src
2   pages
3     page1
4       App.vue
5       index.html
6       page1.js
7   page2
8   page3
9   page4
```

方案一

```
1 src
2   pages
3     page1
4       App.vue
5       index.html
6       index.js
7   page2
8   page3
9   page4
```

方案二

```
1 src
2   pages
3     page1
4       App.vue
5       index.html
6       routes.js
7       index.js
8   page2
9   page3
10  page4
```

方案二 (SPA 单页)

为什么需要多页？

单页 SPA 并行的多页业务场景，不同的个性化业务场景模板（端内、运营、短信营销、PC 等）

如何设计多页？

多入口

webpack.base.conf.js

```
11 module.exports = {  
12   entry: utils.entries(),  
13 }
```

html-webpack-plugin webpack.dev.conf.js

```
].concat(utils.htmlWebpackPlugin())
```

查找 - 对比 **nuxt** 自动路由机制

方案一

```
1 var PAGE_PATH = path.resolve(__dirname, '../src/pages')  
2 var entryFiles = glob.sync(PAGE_PATH + '/*/*.js')
```

方案二

```
1 var PAGE_PATH = path.resolve(__dirname, '../src/pages')  
2 var entryFiles = glob.sync(PAGE_PATH + '/*/*.index.js')
```

Vue.js 多页应用

本地调试越加越慢

Dev 环境下：

读 pages 目录生成 new HtmlWebpackPlugin 数组的时候，加了一个小开关：

```
const IS_DEV_ENV = htmlUtils.isDevEnv()
if (IS_DEV_ENV && debugMode.debug) pages = debugMode.pages
let env = [
```

```
module.exports = {
  debug: true,
  pages: ['test', 'turnplate', 'tool_step1', 'tool_manage']
}
```

Vue.js 多页应用

速度优化 - parallel-uglify

```

    .uglify-cache
    02c9bb19c967df31d39c6d9847b528104ee1da91a3d7942045f00c0090b8ca43.js
    02fe7826e93b23573a7ec2db0f1596bbc42ea5b4c77a399bcfbfd013cf60b2a96.js
    0e8c1f2bba01ac2a8f8a89c0b2ce59ef1eb8eb8848782b4d6aa0d9bd423cdac5.js
    136ba459734640d2a3f05f9636071acbea4f174b55a21cc523179d14fab632a5.js
    13b1edc8d90e77a840b75d57f7e695dbcfc3696ac89e3949e10fbe0980bcb42b.js
    1974e73e96468c998c395c4350174a6a093e86aef4c1779af193d2c2315079d1.js
    1fdc34650590ef2c1769443d4c9e9d249389c68fba075a95ab4eb49e980427bd.js
    2148fd229c92c9dee019272130c2b0f1c9df49486ff269c3f7fad80dea87ac63.js
    2a5d7e47b7e06f2d1260c0d3296976830ab4fc3a2b14f5ff6afed0529ed6d7de.js
    2cb0450e195dd932236966979d65a35b73ee210bd00ccb3e06ff3d8c9f254944.js
    2f3ca0fbd5c9c3e42a964c06c9034f4d5f7a6448bcd7447f5fe9b1e0767de301.js
    31ade07e24152dec0f06259d02838df9dd6e35fb0e86dc1bd5941ee4d7356522.js
    36ab6fb00769b0c3ecc641cb2cdd30466996245489b5be95dd94a3405bf9379d.js
    36ce9366baf7daecb89ce800327bb41e711c02fc8a84f7c3328323cb6da6cb89.js
    38033a1b04884133537139666301fcb5bea10ba8049d9802d41e0f3be4a2ece7.js
    48520c3779bfebaeeab9f4df113eb57744f71e055643305dc4c91e8f3689c76.js
    4890cb7758af33b0b7384b9a6d06c8104fa243f51da6b11ec1f4e13849feb7b6.js
    4bc9e81d37a826d2ee20294c06f8482892800990fea0a3a7d7c0a0df810eb749.js
  
```

	#218976 by	release_tag_0... -> f535a70c faster(paralle)		00:02:10 9 minutes ago
	#218782 by	release_tag_0... -> 3be5cf03 change share icon		00:02:01 about 6 hours ago

我们的多页应用打包规模是：95 个文件

Vue.js 多页应用

速度优化 - parallel-uglify

原理：解决单线程、利用多核

是一个 Webpack 插件

```
1 function FasterUglifyPlugin(options) {}  
2  
3 FasterUglifyPlugin.prototype.apply = function apply(compiler) {  
4  
5 }
```

```
  compiler.plugin('compilation', compilation => {  
    compilation.plugin('optimize-chunk-assets', (chunks, callback) => {  
      uglifier.processAssets(compilation, this.options).then(() => {  
        callback();  
      });  
    });  
  });
```

```
  compiler.plugin('done', () => {  
    uglifier.pruneCache(this.options);  
  });
```

```
└─ webpack-parallel-uglify-plugin@1.1.0  
  └─ babel-code-frame@6.26.0 deduped  
    └─ glob@7.1.2 deduped  
      └─ mkdirp@0.5.1 deduped  
        └─ pify@3.0.0  
          └─ tmp@0.0.29  
            └─ os-tmpdir@1.0.2 deduped  
              └─ uglify-es@3.3.9  
                └─ commander@2.13.0  
                  └─ source-map@0.6.1 deduped  
                    └─ uglify-js@3.1.8  
                      └─ commander@2.11.0 deduped  
                        └─ source-map@0.6.1 deduped  
                          └─ webpack-sources@1.0.2 deduped  
                            └─ worker-farm@1.6.0  
                              └─ errno@0.1.7  
                                └─ prr@1.0.1
```

Vue.js 多页应用

速度优化 - parallel-uglify 不明显的原因

- 1、runner 不固定
- 2、会先做 Removing 操作

```
Running with gitlab-runner 10.0.2 (a9a76a50)
  on hk-runner-fe-2 (9898940a)
Using Shell executor...
Running on hk-runner-fe-2...
Fetching changes...
Removing .uglify-cache/
Removing dist/
Removing node_modules/
HEAD is now at 320e602 xxxxxx
From https://xxxxxx
  3b785b4..c0a20d0  release/730 -> origin/release/730
* [new tag]          dev_tag_04171048 -> dev_tag_04171048
Checking out c0a20d06 as dev_tag_04171048...
Skipping Git submodules setup
Checking cache for default...
Successfully extracted cache
```

Vue.js 多页应用

打包 - Vendor.js 文件臃肿

1、大在哪里？

webpack-bundle-analyzer

config 里面

2、环境？

webpack.prod.conf.js 文件

```
// Set to `true` or `false` to always turn it on or off  
bundleAnalyzerReport: process.env.npm_config_report
```

```
if (config.build.bundleAnalyzerReport) {  
  const BundleAnalyzerPlugin = require('webpack-bundle-analyzer').BundleAnalyzer  
  webpackConfig.plugins.push(new BundleAnalyzerPlugin())  
}
```

打包 - Vendor.js 文件臃肿

1、拆除

```
const webpackConfig = merge(baseWebpackConfig, {  
  module: {  
    rules: utils.styleLoaders({  
      sourceMap: config.build.productionSourceMap,  
      extract: true  
    })  
  },  
  externals: {  
    'vue': 'Vue'  
  },  
  devtool: config.build.productionSourceMap ? '#source-map' : false,  
  output: {  
    path: config.build.outputPath,  
    filename: config.build.outputFilename,  
    chunkFilename: config.build.chunkFilename  
  }  
})
```

2、引入

```
<link rel="dns-prefetch" href="https://2-static.mobike.com"/>  
<% if(htmlWebpackPlugin.options.env == 'production') { %>  
<script src="https://static.mobike.com/cdn/vue/2.5.13/vue.min.js"></script>  
<% } %>  
<title>activity-vue</title>  
</head>
```

打包 - Vendor.js 文件臃肿

1、拆除前后

```
1 webpackJsonp([76], {
2   "0duY": function(t, n, e) {=
23   },
24   BHUn: function(t, n, e) {=
50   },
51   "PtB+": function(t, n) {=
62   },
63   Un3m: function(t, n) {},
64   kQ3z: function(t, n, e) {=
69   },
70   m6D0: function(t, n, e) {=
113  },
114  w8ZG: function(t, n, e) {=
145  }
146 }, ["BHUn"]);
```

```
1 webpackJsonp([75], {
2   0: function(t, n) {
3     t.exports = Vue
4   },
5   "0duY": function(t, n, e) {=
26   },
27   BHUn: function(t, n, e) {=
53   },
54   "PtB+": function(t, n) {=
65   },
66   Un3m: function(t, n) {},
67   kQ3z: function(t, n, e) {=
72   },
73   m6D0: function(t, n, e) {=
116  },
117  w8ZG: function(t, n, e) {=
148  }
149 }, ["BHUn"]);
```

```
(function handleExternals(externals, callback) {
  console.log(arguments)
  if(typeof externals === "string") {
  } else if(Array.isArray(externals)) {
  } else if(externals instanceof RegExp) {
  } else if(typeof externals === "function") {
  } else if(typeof externals === "object" && Object.
    return handleExternal(externals[dependency.rec
  }
  callback();
})(this.externals, function(err, module) {
  if(err) return callback(err);
  if(!module) return handleExternal(false, callback)
  return callback(null, module);
}));
```

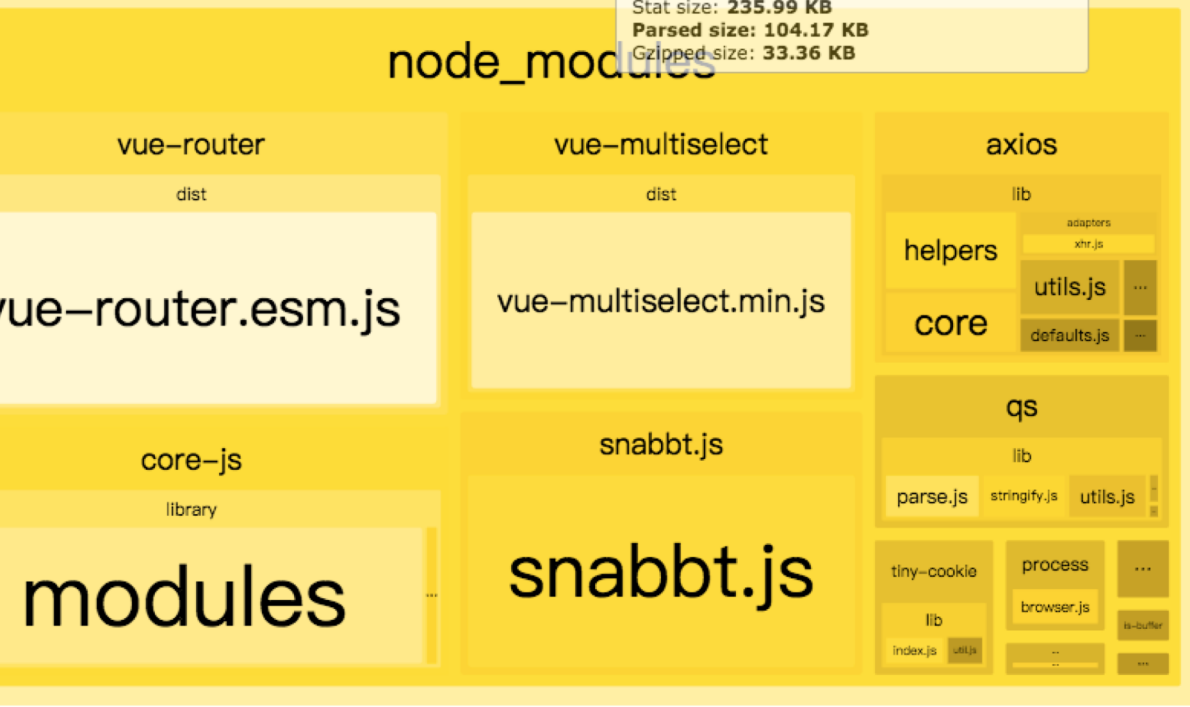
static/js/vendor.007609b20502294b55d1.js

Stat size: 182.42 KB
Parsed size: 69.95 KB
Gzipped size: 23.84 KB

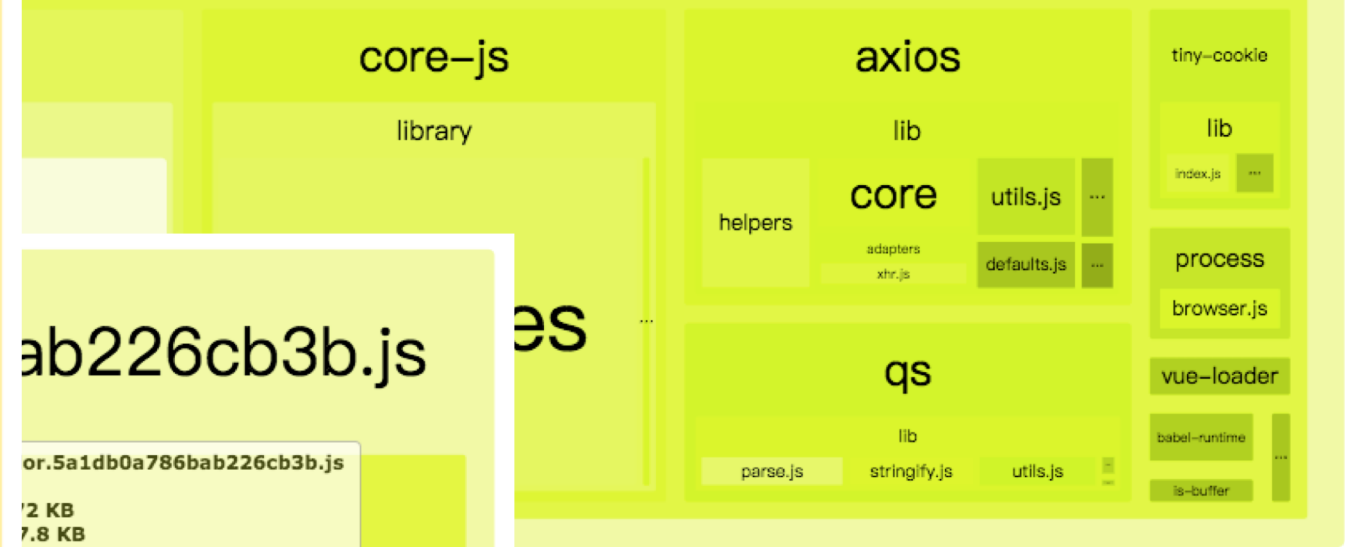


static/js/vendor.c30309fcf2be618ecf41.js

static/js/vendor.c30309fcf2be618ecf41.js
Stat size: 235.99 KB
Parsed size: 104.17 KB
Gzipped size: 33.36 KB



node_modules



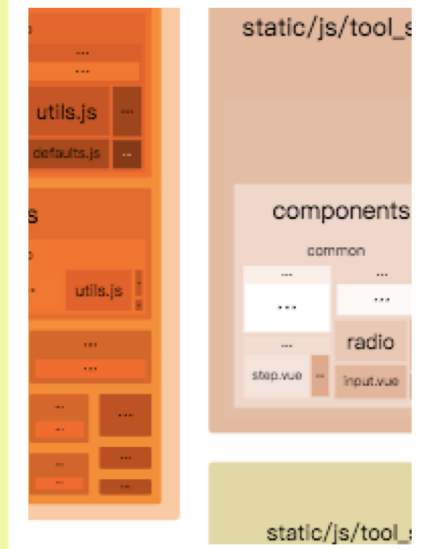
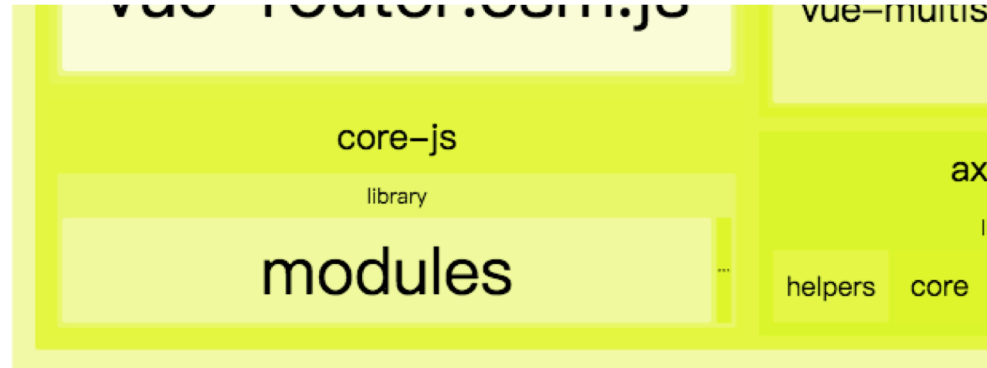
786bab226cb3b.js

or.5a1db0a786bab226cb3b.js
2 KB
7.8 KB
3.5 KB



vue-router.esm.js

vue-multiselect.min.js



static/js/vendor.239c06d002d6c9f140c6.js

static/js/vendor.239c06d002d6c9f140c6.js
Stat size: **504.19 KB**
Parsed size: **190.7 KB**
Gzipped size: **66.81 KB**

Show

- All
- sta
- sta
- sta
- sta
- sta
- sta
- sta
- sta
- sta
- sta

node_modules

vue
dist

vue.js

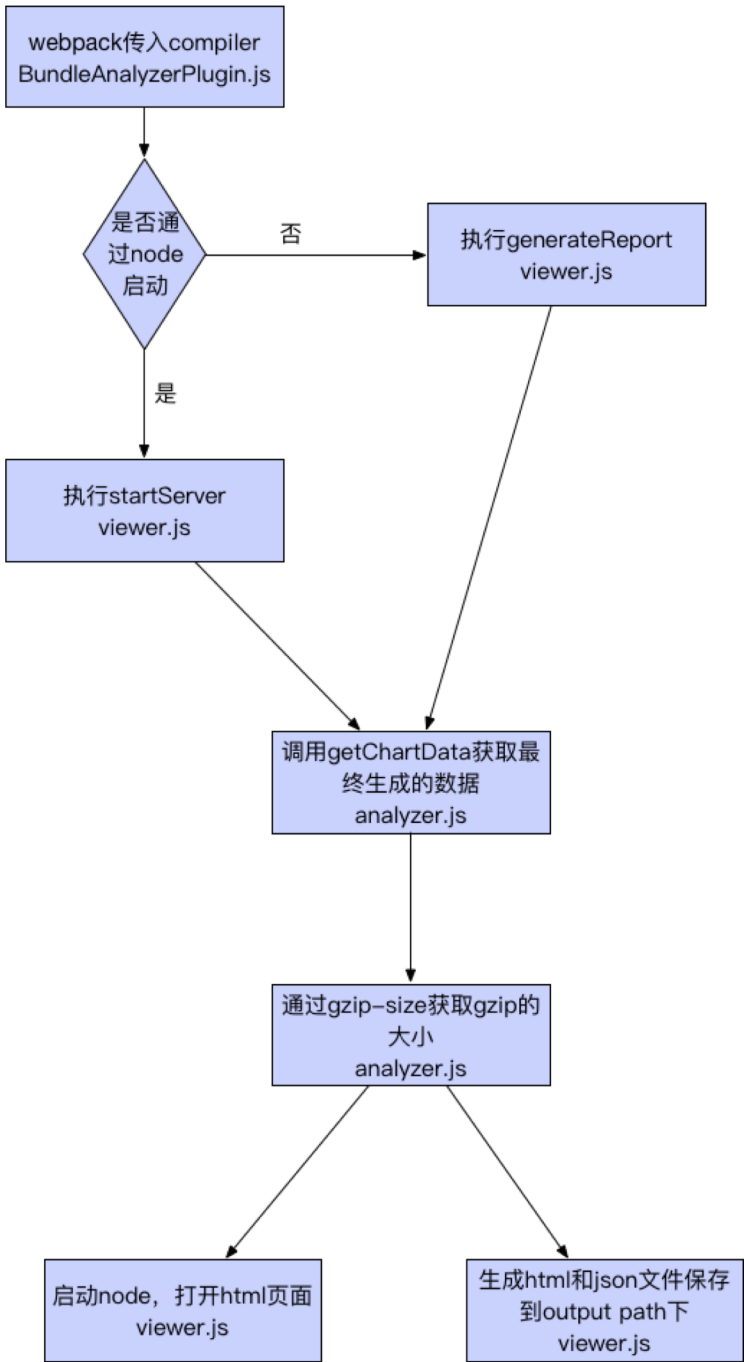


static/js/vendor.fba8685e3dedbe176b7e.js

static/js/vendor.fba8685e3dedbe176b7e.js
Stat size: **222.15 KB**
Parsed size: **84.74 KB**
Gzipped size: **28.16 KB**



36.02 KB)
(KB)
5.js (29.97 KB)
6.js (29.7 KB)
n.esm.js
js (21.31 KB)
KB)
KB)
s (18.78 KB)



```

1 class BundleAnalyzerPlugin {
2   constructor(opts) {}
3   apply(compiler) {}
4   async generateStatsFile(stats) {}
5   async startAnalyzerServer(stats) {}
6   generateStaticReport(stats) {}
7   getBundleDirFromCompiler() {}
8 }
  
```

```

compiler.plugin('done', done);
  
```


bundle 信息获取的方式：

bundleDir → compiler.outputPath

```
{ errors: [],
  warnings: [],
  publicPath: 'https://static.mobike.com/active/',
  assetsByChunkName: {},
  assets: [Array],
  entrypoints: [Object],
  chunks: [Array],
  modules: [Array],
  filteredModules: 0,
  children: [],
  name: 'html-webpack-plugin for "game/index.html" },
  ... }
```

```
bundleStats.assets [ { name: 'static/js/mobi_invite.81e7f16c3d69385c4f7b.js',
  size: 37465,
  chunks: [ 53 ],
  chunkNames: [ 'mobi_invite' ],
  emitted: true,
  isOverSizeLimit: undefined },
  { name: 'static/img/background.a3a382e.png',
  size: 52876,
  chunks: [],
  chunkNames: [],
  emitted: true,
  isOverSizeLimit: undefined },
  ... ]
```

/Users/zhangyaochun/***/dist/active/static/js/mobi_invite.81e7f16c3d69385c4f7b.js

```
for (const statAsset of bundleStats.assets) {
  const assetFile = path.join(bundleDir, statAsset.name);
  console.log('assetFile', assetFile)
  let bundleInfo;

  try {
    bundleInfo = parseBundle(assetFile);
  } catch (err) {
    bundleInfo = null;
  }
}
```

```
const content = fs.readFileSync(bundlePath, 'utf8');
const ast = acorn.parse(content, {
  sourceType: 'script',
  ecmaVersion: 2015
});
```

DevOps 篇

小公司里，一个团队可能负责整个开发生命周期

在大公司里，是被各个部门承担的

包含什么？

1、规划

Confluence

2、构建

3、持续集成

4、自动化部署

5、运营

应用程序及服务器性能监控

事件、变更和问题跟踪

6、持续反馈

DevOps 篇 - 私服

1、npm

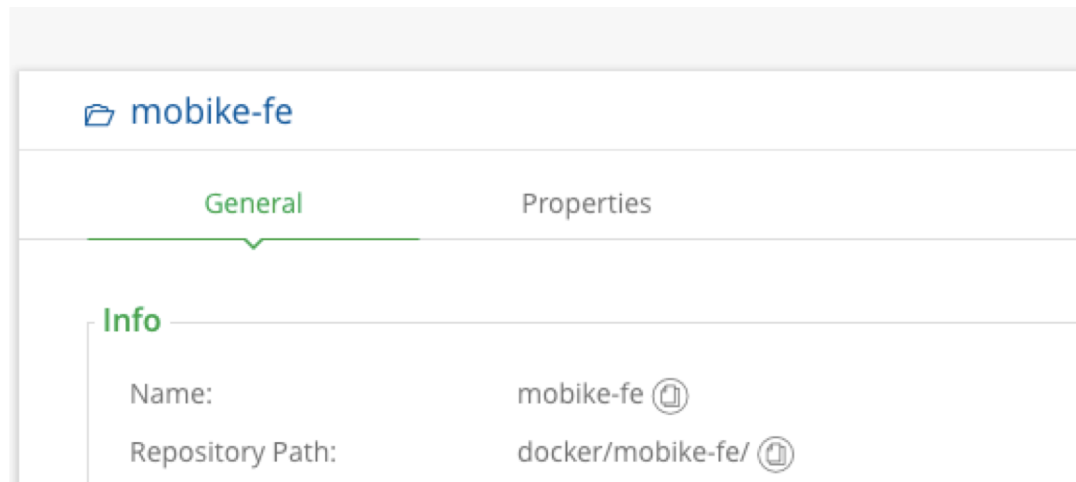
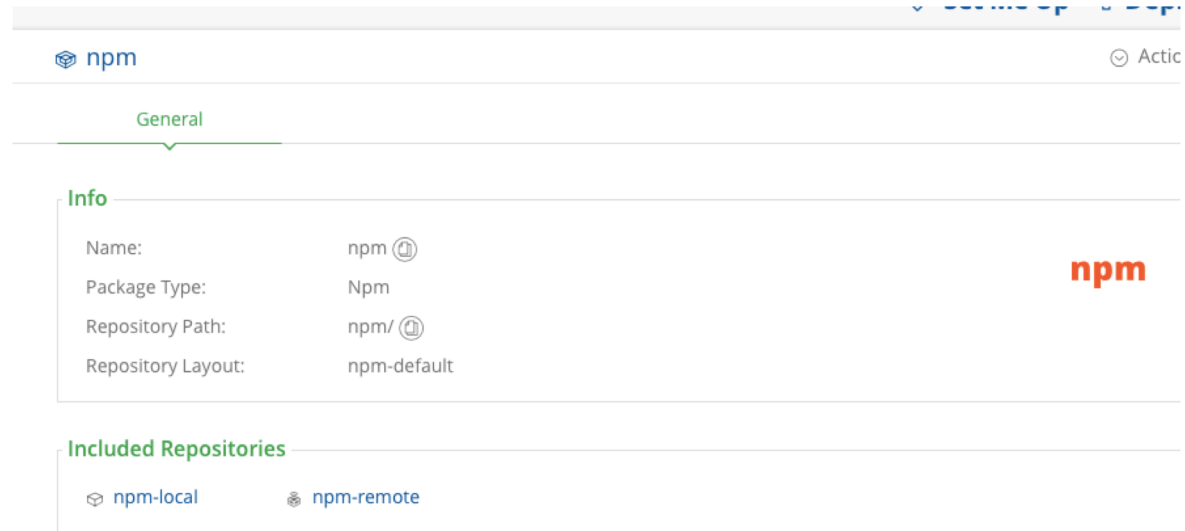
npm init --scope=mobike

2、docker

docker tag

docker push

3、node 多版本镜像



DevOps 篇 - 发布

从 make tag 开始

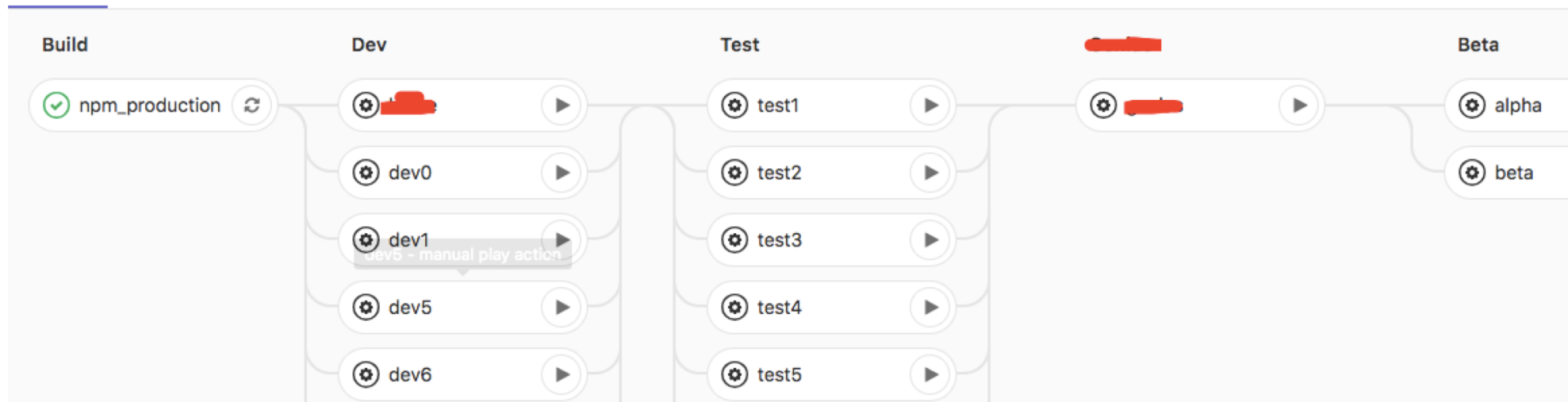
- Overview
- Repository
- Issues 0
- Merge Requests 2
- CI / CD
- Pipelines

All **2693**
Pending **0**
Running **0**
Finished **2693**
Branches
Tags

Run Pipeline
CI Lint

Status	Pipeline	Commit	Stages	Duration	
passed	#222634 by latest	release_tag_0... -> 9ef22ca3 	✔ » » » »	⌚ 00:07:10 📅 about 11 hours ago	▶
passed	#221945 by latest	dev_tag_04201... -> f2c5bb81 	✔ » » » »	⌚ 00:07:00 📅 about 22 hours ago	▶

Pipeline Jobs **21**



DevOps 篇 - 发布

做了什么？

Running runner

Checking out 9ef22ca3 as tag 名字

npm install

rm -fr dist

npm run deploy

docker build

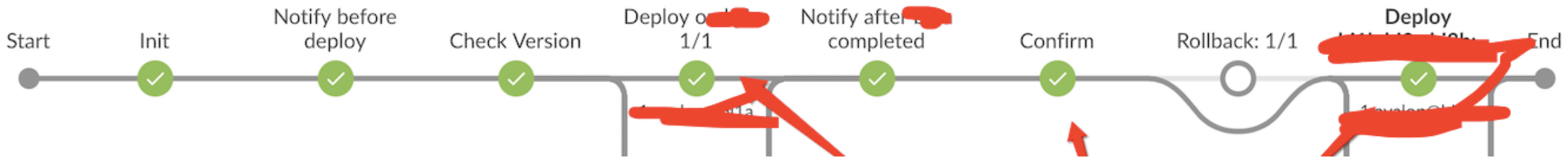
docker tag

docker push 一般会有 2 次！！！！

Dockerfile

```
1 FROM docker.*/infra/**-node:node8
2 RUN mkdir -p /**
3 ENV CONSUL_TAGS fe,**
4 COPY . /**
5 WORKDIR /**
6 EXPOSE **
7 RUN npm install
8 RUN npm run build
9 CMD ["start"]
```

前端国际化发布



docker-compose.yml

docker-compose up

ansible

1、ansible – 基于 Python 开发、模块工作

2、ansible-playbook

命令

- --user
-u REMOTE_USER, --user=REMOTE_USER
connect as this user (default=None)
- -m
-m MODULE_NAME, --module-name=MODULE_NAME
module name to execute (default=command)
- -i
-i INVENTORY, --inventory=INVENTORY, --inventory-file=INVENTORY
specify inventory host path
(default=[[u'/etc/ansible/hosts']]) or comma separated
host list. --inventory-file is deprecated
- --extra-vars
-e EXTRA_VARS, --extra-vars=EXTRA_VARS
set additional variables as key=value or YAML/JSON, if
files are specified with @

ansible

模块

- copy 拷贝模块

ansible ***** -m copy -a "src=***** dest=*****"

- shell 核心的神器模块

ansible ***** -m shell -a "cd *****"

➔ 这个就可以做类似 docker 的操作了

yml – 配置文件

数据结构

- 对象
- 数组
- 复合结构
- 纯量

```
1  npm:
2      stage: build
3      script:
4      - ***
5      only:
6      - ***
```

自动化测试 – 验证海外节点页面正常

核心思路：

利用 python + webdriver selenium

操作流程：

安装：

```
yum -y install python-pip  
pip install selenium
```

注释：pip 类似 yum，安装 python 包 ----- Performance Improvement Plan, 即绩效改进计划

核心方法：

```
browser = webdriver.Firefox()  
browser.maximize_window()  
browser.implicitly_wait(6)  
browser.get(url)  
browser.save_screenshot(path)  
browser.quit()
```

前端国际化

多页 + 国际化

编译之后的目录结构

所以是一个 $n * k$

n - page 的个数

k - 语言的个数

```
1 dist
2   h5
3     accout
4       zh
5         index.html
6       zh-tw
7     en
8     de
9     th
10    nl
11    ms
12    mr
13    ko
14    ja
15    it
16    fr
17    es
18    es-cl
19    es-mx
```

SSR 实践

1、nginx 静态化的问题

如果出现了一个打包目录没有的语言前缀，如何 feedback 到 en

2、服务化

```
*****/about/:lang/index.html
```

路由匹配规则 (vue-router routes) + 一份文件 (业务和国际化资源包)

Nuxt i18n

```
1 pages
2   _lang
3   about.vue
```

1、路由问题需要匹配之前的国际化规则

```
1 dist
2   h5
3     accout
4     zh
5     index.html
6     zh-tw
7     en
8     de
9     th
10    nl
11    ms
12    mr
13    ko
14    ja
15    it
16    fr
17    es
18    es-cl
19    es-mx
```

编译之后的目录结构

所以是一个 $n * k$

n – page 的个数

k – 语言的个数

.nuxt 目录下面 router.js

```
import Vue from 'vue'
import Router from 'vue-router'
```

```
Vue.use(Router)
```

```
const _51d31151 = () => import('../pages/_lang/about.vue'
/* webpackChunkName: "pages/_lang/about" */).
then(m => m.default || m)
```

```
{
  path: "/:lang/about",
  component: _51d31151,
  name: "lang-about"
}
```

Nuxt i18n

我们需要一个：

1、智能版（懒惰版）：

```
1 about/zh/index.html
2
3 about
4   _lang
5   index.html(文件夹)
```

2、升级版（可扩展性）：

```
1 router: {
2   middleware: 'i18n',
3   extendRoutes (routes) {
4     // ...
5   }
6 }
```

遇到最大的问题：

i18n 的 messages 如何**按需加载**

Nuxt i18n

核心设计：

```
1 router: {  
2   middleware: 'i18n'  
3 },  
4 plugins: [  
5   {src: '~plugins/i18n'}  
6 ]
```

plugins/i18n.js

```
1 import Vue from 'vue'  
2 import VueI18n from 'mbk-vue-i18n'  
3  
4 Vue.use(VueI18n)  
5  
6 export default ({ app, isClient, store }) => {  
7   app.i18n = new VueI18n({  
8     locale: store.state.locale,  
9     fallbackLocale: 'en'  
10  })  
11 }
```

middleware/i18n.js

Egg vue SSR 实践

```
1 app
2   controller
3   extend
4   middleware
5   service
6   view
7   web
8   router.js
9 config
10  config.default.js
11  config.local.js
12 logs
13 public
14 run
15 index.js
16 webpack.config.js
17 webpack.prod.config.js
```

1、前后端渲染layout

*前端渲染layout通过config配置vuessr插件指定

*后端渲染layout通过web/framework/vue/component/index.js指定

2、统计类、flexible 注入类脚本

*不能直接在 layout 写

*配置的方式

```
exports.vuessr = {
  layout: '*****',
  injectRes: [{
    inline: true,
    location: 'headBefore',
    url: path.join(app.baseDir, 'app/web/utils/flexible.js')
  },
  {
    inline: true,
    location: 'bodyAfter',
    url: path.join(app.baseDir, 'app/web/utils/analytic.js')
  }
];
```

Egg vue ssr 实践

1、引入 node_modules 下 vue 组件报找不到对应的 loader 错误

<https://github.com/hubcarl/egg-vue-webpack-boilerplate/issues/53>

include 配置

例如：代码在 app/web 目录，引入 node_modules 下 vue 组件为

```
// ${app_root}/webpack.config.js
module.exports= {
  loaders :{
    vue: {
      include: [/app\web/, /node_module\vue-datepicker-local/]
    }
  }
};
```

相关issue: [import 外部模块失败](#)

exclude 配置

```
// ${app_root}/webpack.config.js
module.exports= {
  loaders :{
    vue: {
      exclude: []
    }
  }
};
```

Egg vue ssr 实践

2、后端渲染：配置vuex store报错问题解决

<https://github.com/hubcarl/egg-vue-webpack-boilerplate/issues/41>

```
import Vue from 'vue';
import '../filter';
import '../directive';
import '../component';
|
export default function(options) {
  Vue.prototype.$http = require('axios');
  if (options.store) {
    options.store.replaceState(window.__INITIAL_STATE__ || {});
  } else if (window.__INITIAL_STATE__) {
    options.data = Object.assign(window.__INITIAL_STATE__, options.data && options.data);
  }
  const app = new Vue(options);
  app.$mount('#app');
}

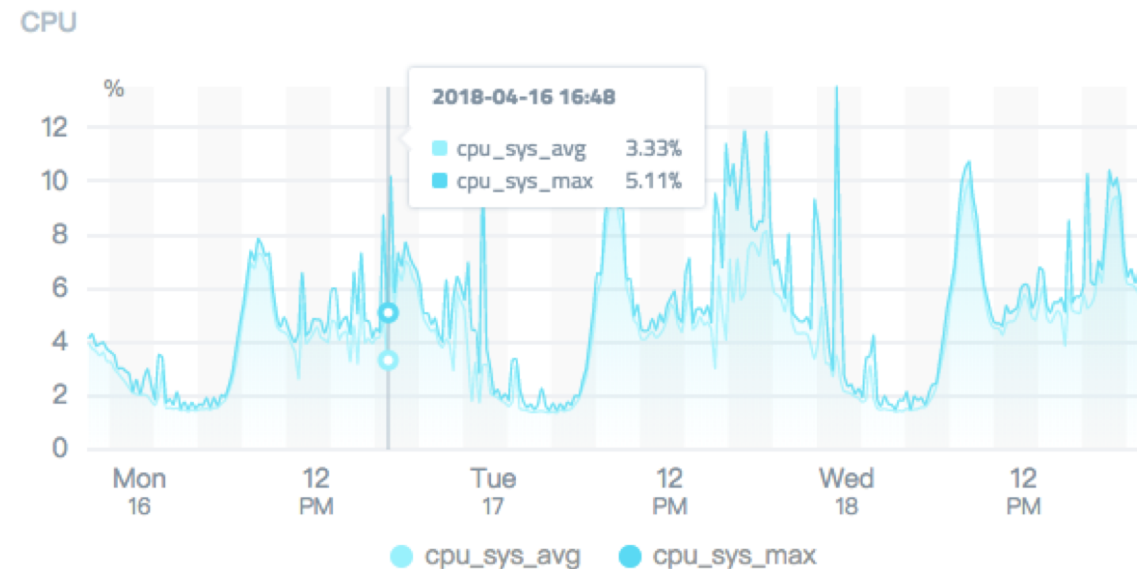
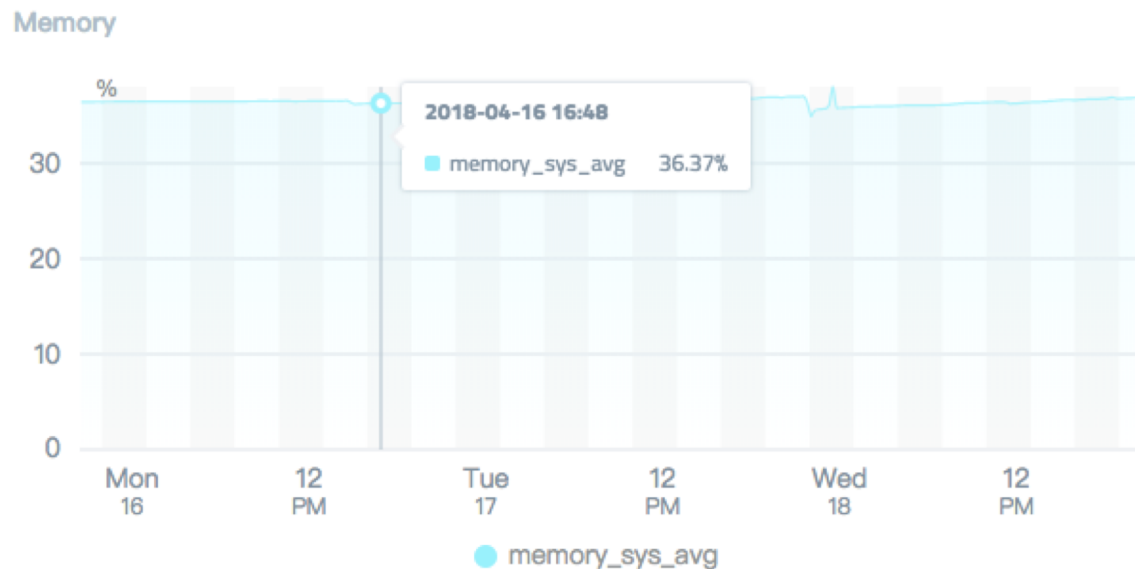
import Vue from 'vue';
import '../filter';
import '../directive';
import '../component';
|
export default function(options) {
  // Vue.prototype.$http = require('axios');
  if (options.store) {
    options.store.replaceState(Object.assign({}, window.__INITIAL_STATE__, options.data));
  } else if (window.__INITIAL_STATE__) {
    options.data = Object.assign(window.__INITIAL_STATE__, options.data && options.data);
  }
  const app = new Vue(options);
  app.$mount('#app');
}
```

Alinode 接入监控

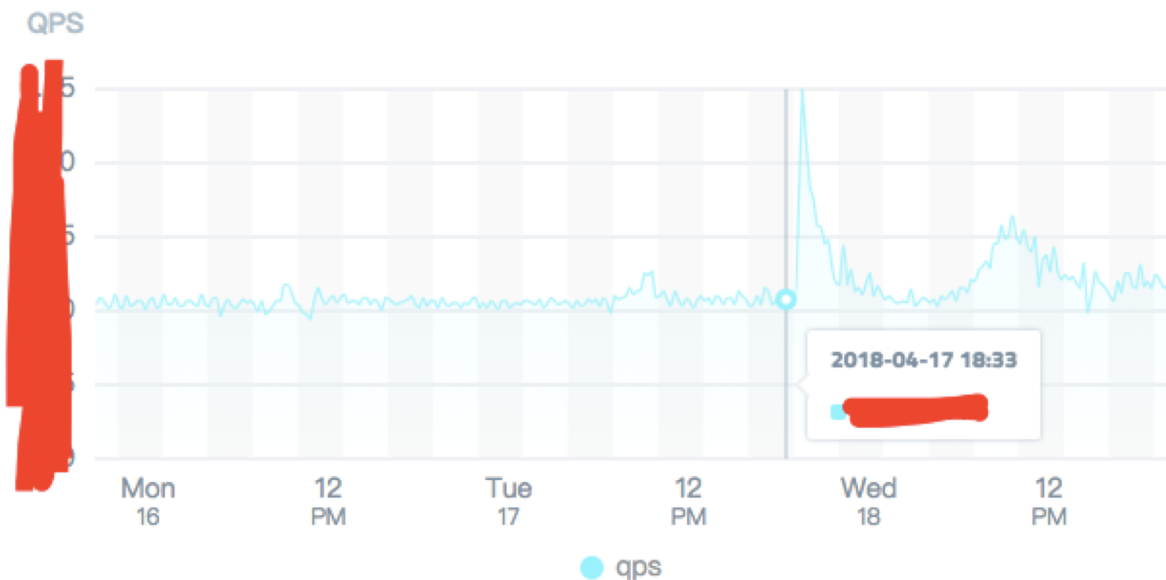
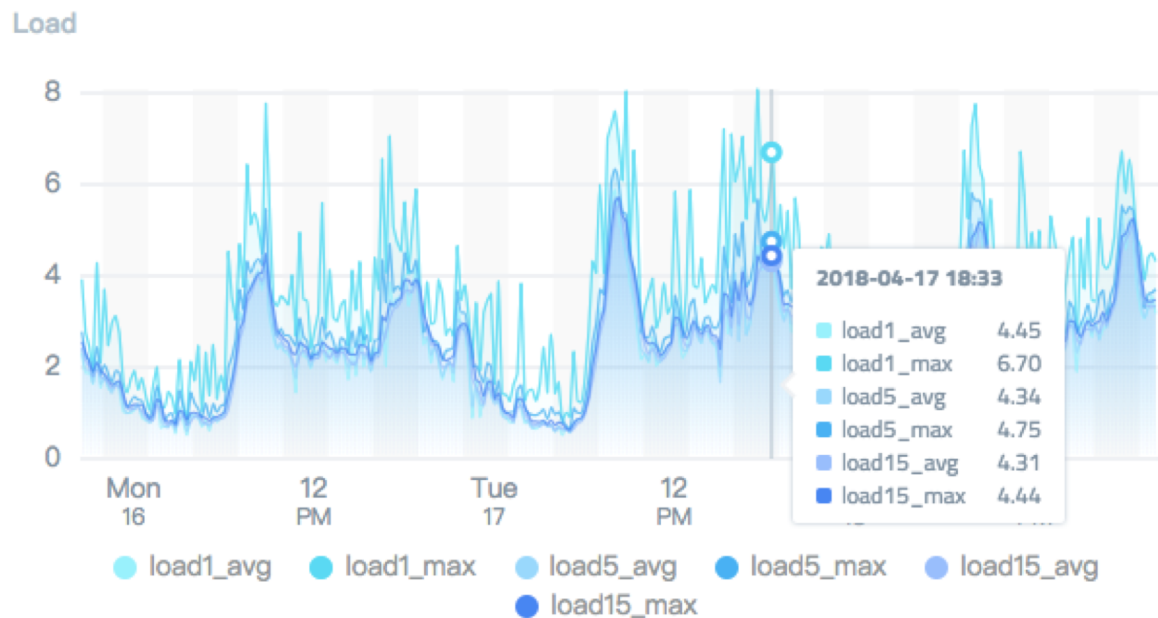
问题：

- 1、容器化部署方式：
配置 ip 来防止名字不变导致的重复
- 2、多个 appid 来区分环境

```
1 exports.alinode = {  
2   appid: '**',  
3   secret: '****',  
4   agentidMode: 'IP'  
5 };
```



Egg vue ssr 实践



核心还是：[agentx](#)

系统中存在以下 Node.js 进程（不一定由 alinode 启动），可以通过[点击检查进程](#)来查看详情

PID	命令	操作
29	node /site/node_modules/.bin/cross-env EGG_SERVER_ENV=prod NODE_ENV=production egg-scripts start --port=3000	检查进程
35	node /site/node_modules/.bin/egg-scripts start --port=3000	检查进程
45	node /site/node_modules/egg-scripts/lib/start-cluster {"port":3000,"title":"egg-server-site-ssr","framework":"/site/node_modules/egg","baseDir":"/site"} --title=egg-server-site-ssr	检查进程
55	/site/node_modules/node/bin/node /site/node_modules/egg-cluster/lib/agent_worker.js {"framework":"/site/node_modules/egg","baseDir":"/site","port":3000,"workers":48,"plugins":null,"https":false,"key":"","cert":"","typescript":false,"title":"egg-server-site-ssr","clusterPort":46631}	检查进程
67	/site/node_modules/node/bin/node /site/node_modules/egg-cluster/lib/app_worker.js {"framework":"/site/node_modules/egg","baseDir":"/site","port":3000,"workers":48,"plugins":null,"https":false,"key":"","cert":"","typescript":false,"title":"egg-server-site-ssr","clusterPort":46631}	检查进程
71	/site/node_modules/node/bin/node /site/node_modules/egg-cluster/lib/app_worker.js {"framework":"/site/node_modules/egg","baseDir":"/site","port":3000,"workers":48,"plugins":null,"https":false,"key":"","cert":"","typescript":false,"title":"egg-server-site-ssr","clusterPort":46631}	检查进程
73	/site/node_modules/node/bin/node /site/node_modules/egg-cluster/lib/app_worker.js {"framework":"/site/node_modules/egg","baseDir":"/site","port":3000,"workers":48,"plugins":null,"https":false,"key":"","cert":"","typescript":false,"title":"egg-server-site-ssr","clusterPort":46631}	检查进程
82	/site/node_modules/node/bin/node /site/node_modules/egg-cluster/lib/app_worker.js {"framework":"/site/node_modules/egg","baseDir":"/site","port":3000,"workers":48,"plugins":null,"https":false,"key":"","cert":"","typescript":false,"title":"egg-server-site-ssr","clusterPort":46631}	检查进程

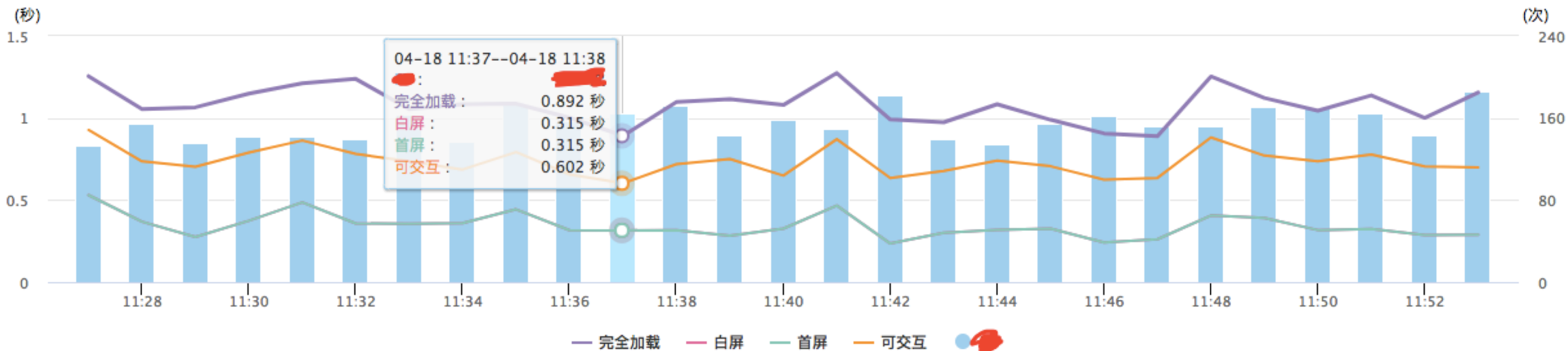
Mobike Node.js 服务探活监控

CHECKS

CHECKS	
Serf Health Status serfHealth	passing
NOTES	
OUTPUT	Agent alive and reachable

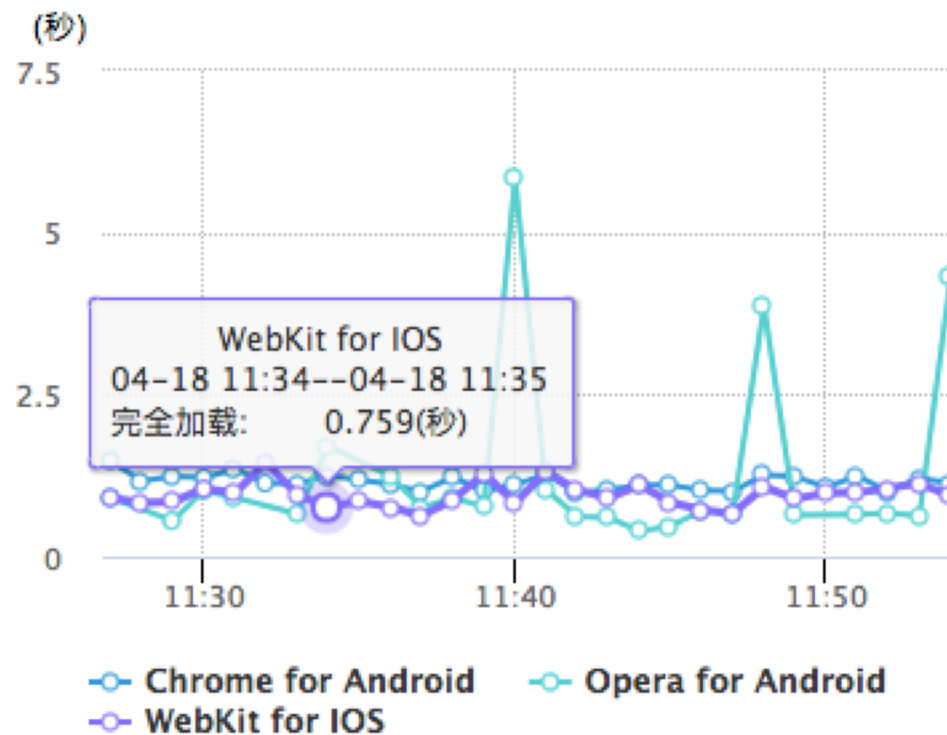
CHECKS	
Version service: [REDACTED]	passing
NOTES	
OUTPUT	

性能方向



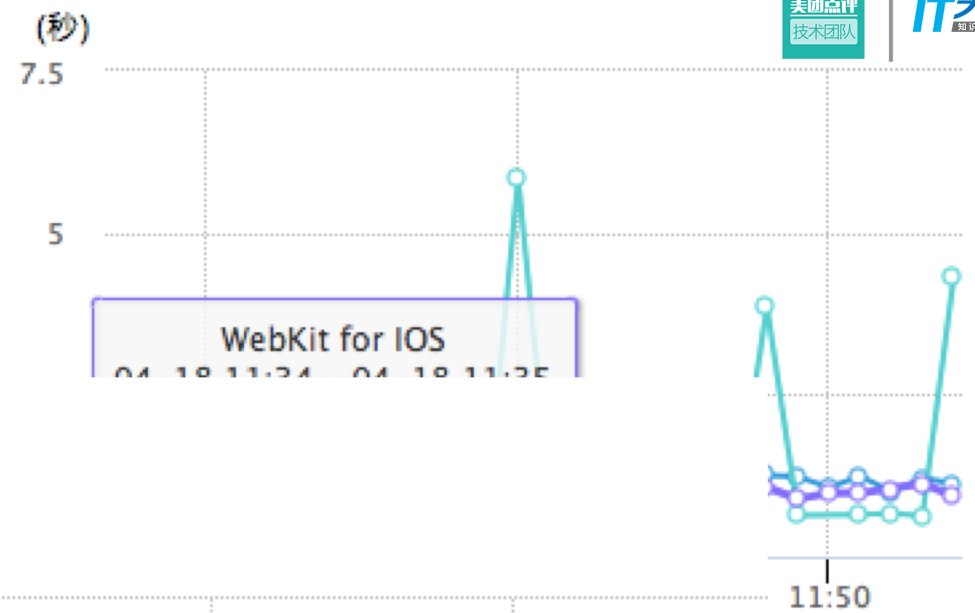
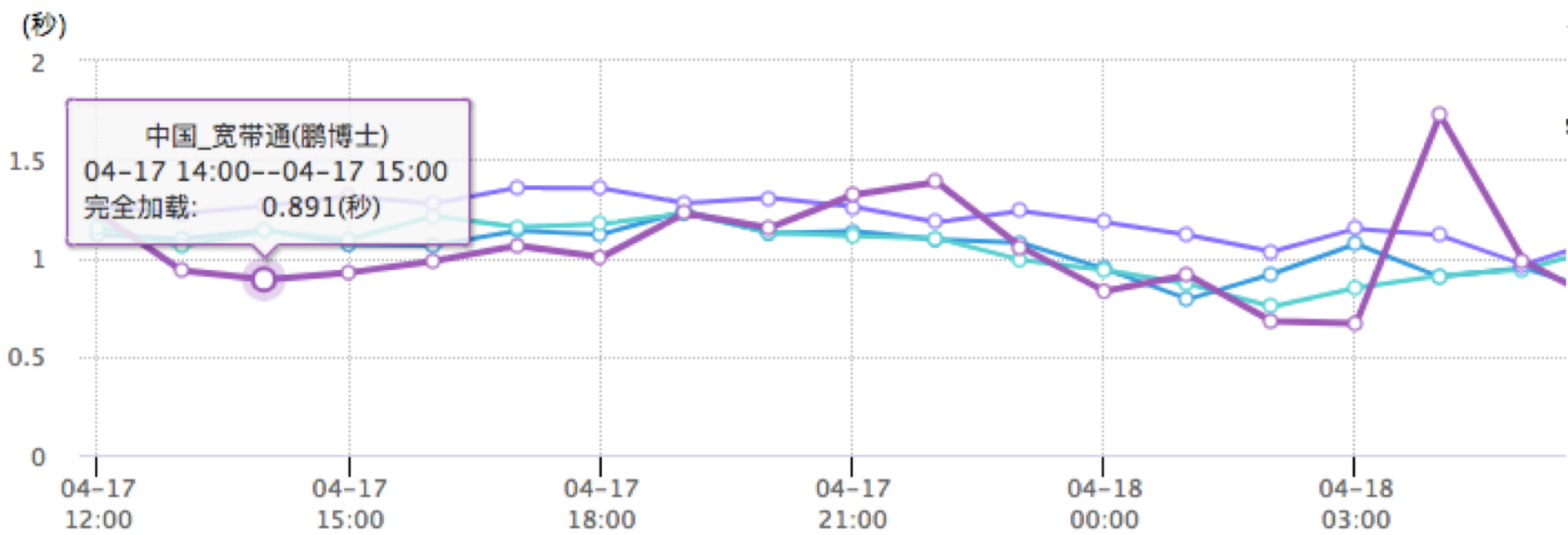
完全加载(s) ↓	白屏(s) ↓	首屏(s) ↓	可交互(s) ↓	JS错误率(%) ↓	慢页面占比(%) ↓	慢页面次数(次) ↓
1.155	0.377	0.377	0.785	0.88	2.31	3968
1.249	0.203	0.203	0.788	0.44	2.71	86
1.384	0.229	0.229	0.974	0.72	3.60	5
1.181	0.263	0.263	0.813	0	1.41	1

性能方向



运营商页面耗时 ?

- 2800ms以上
- 2400-2800ms
- 2000-2400ms
- 1600-2000ms
- 1200-1600ms
- 800-1200ms
- 400-800ms
- 0-400ms



Android

中国_中国联通 中国_中国电信 中国_中国移动 中国_宽带通(鹏博士)

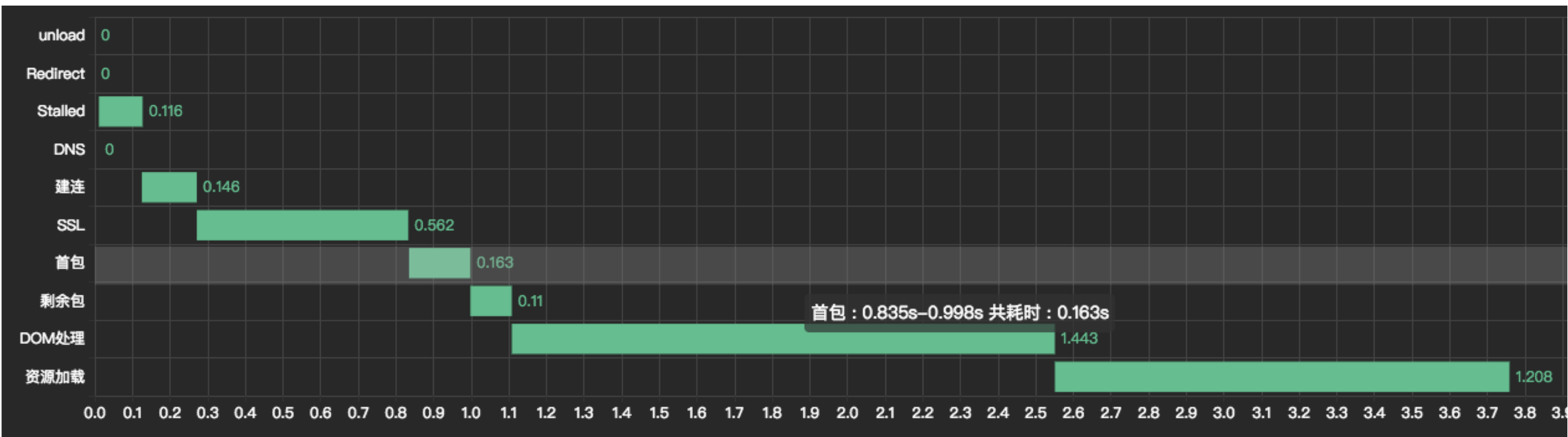
性能方向

错误类型	JS文件名	行	列	浏览器
Uncaught TypeError: Cannot read property '1' of null	#	1	69	WebKit for Android-未知

UA
数
据

HUAWEI TAG-TL00/1.0 Linux/3.10.65 Android/5.1 Release/07.10.2015 Browser/AppleWebKit537.36 Chrome/39.0.0.0 Mobile Safari/537.36 Mobike/7.1.0 X-Tingyun-Id/2wQNFaevdJI;c=2;r=806344422;

性能



白屏

首屏

可交互

自定义加载



2.228s

2.228s

2.533s

0ms



请求量(个)

13

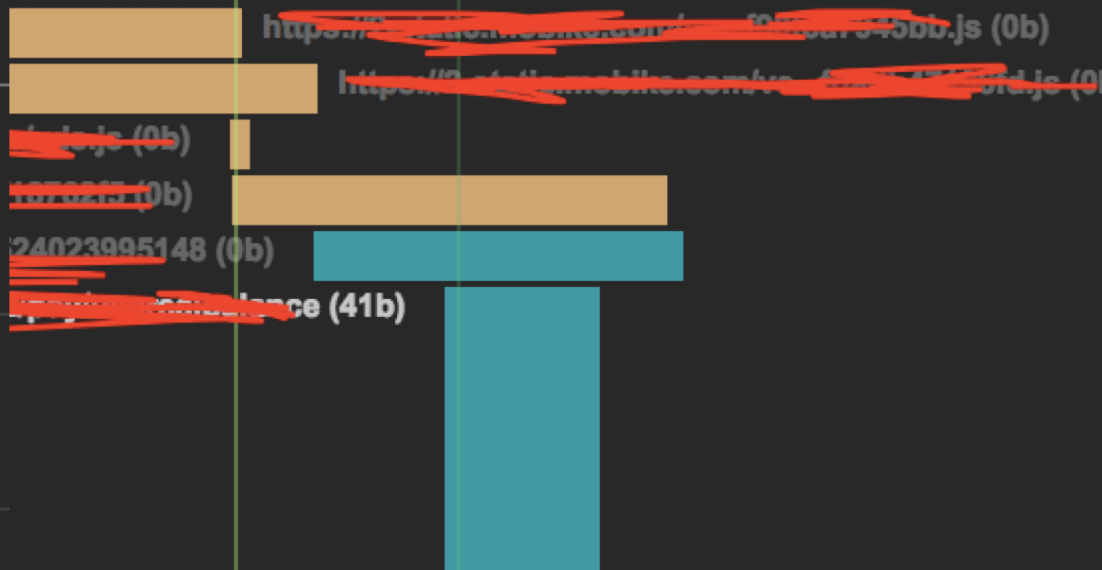
15

10

5

0

script: 5↑



可用性



出错了，您访问的页面无法找到...

活动已结束
我们下期再会~



更多内容 查看APP

mobike
摩拜单车



这个城市暂未开通摩拜单车业务
敬请期待，感谢您的关注~

.....

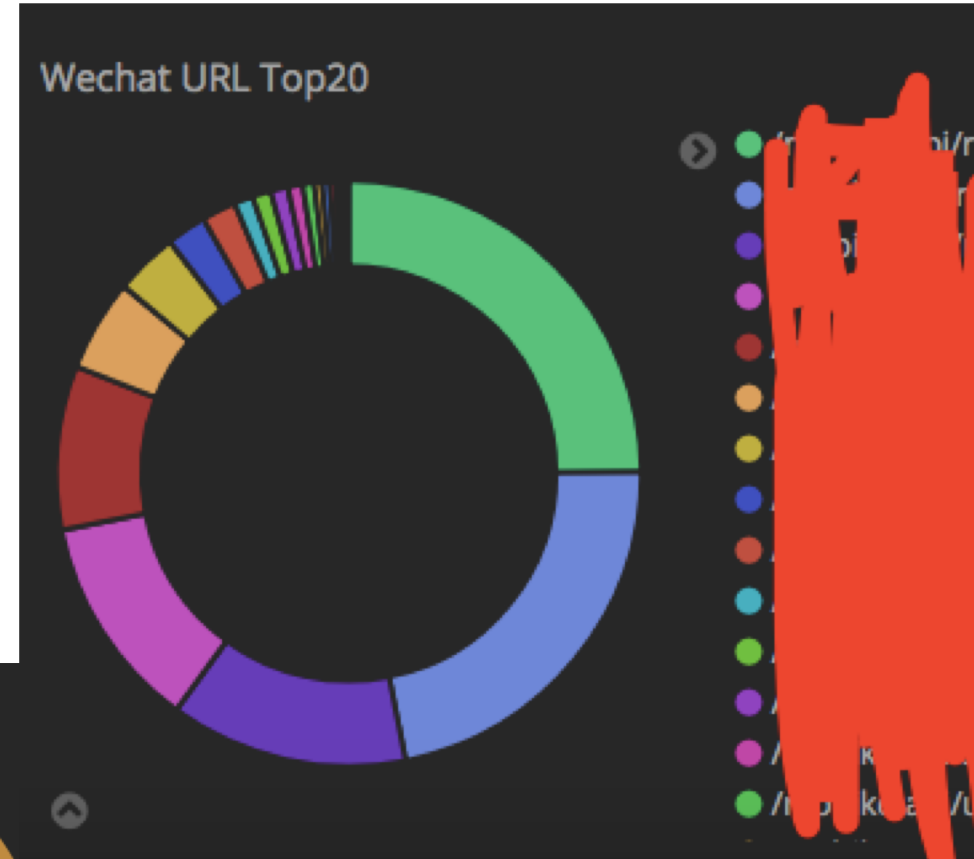
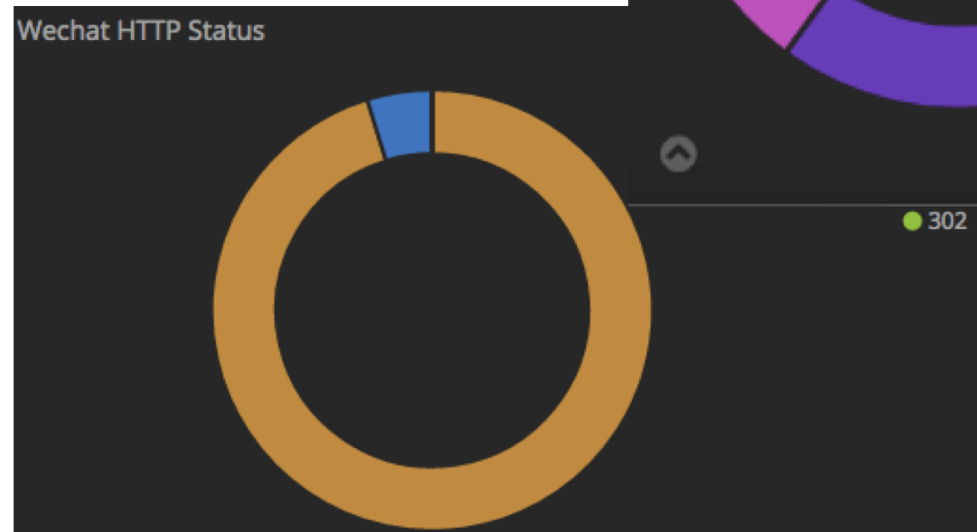
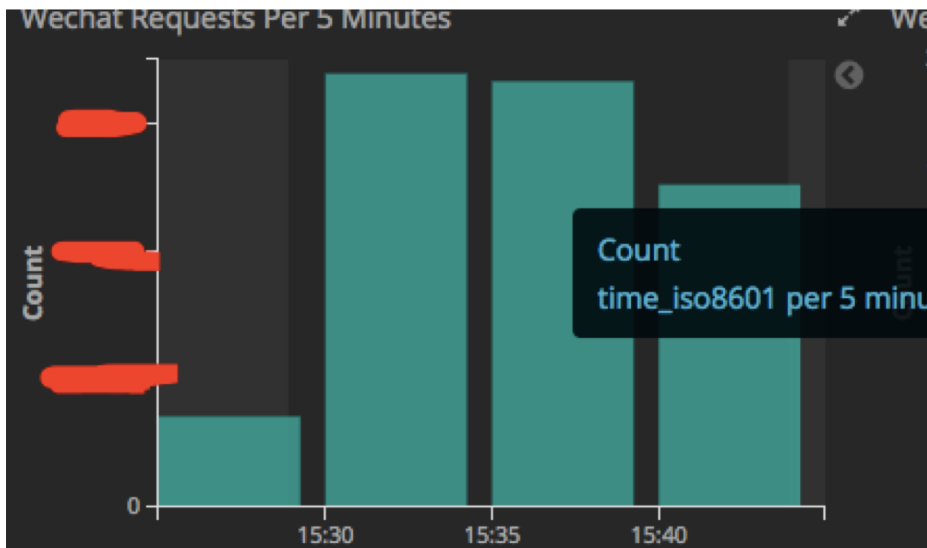
可用性

1、所有集群的 404 监控

因为容器化带来的问题

2、可视化 dashboard

基于 kibana -- 前端用的 angular 写的







摩拜前端的特色

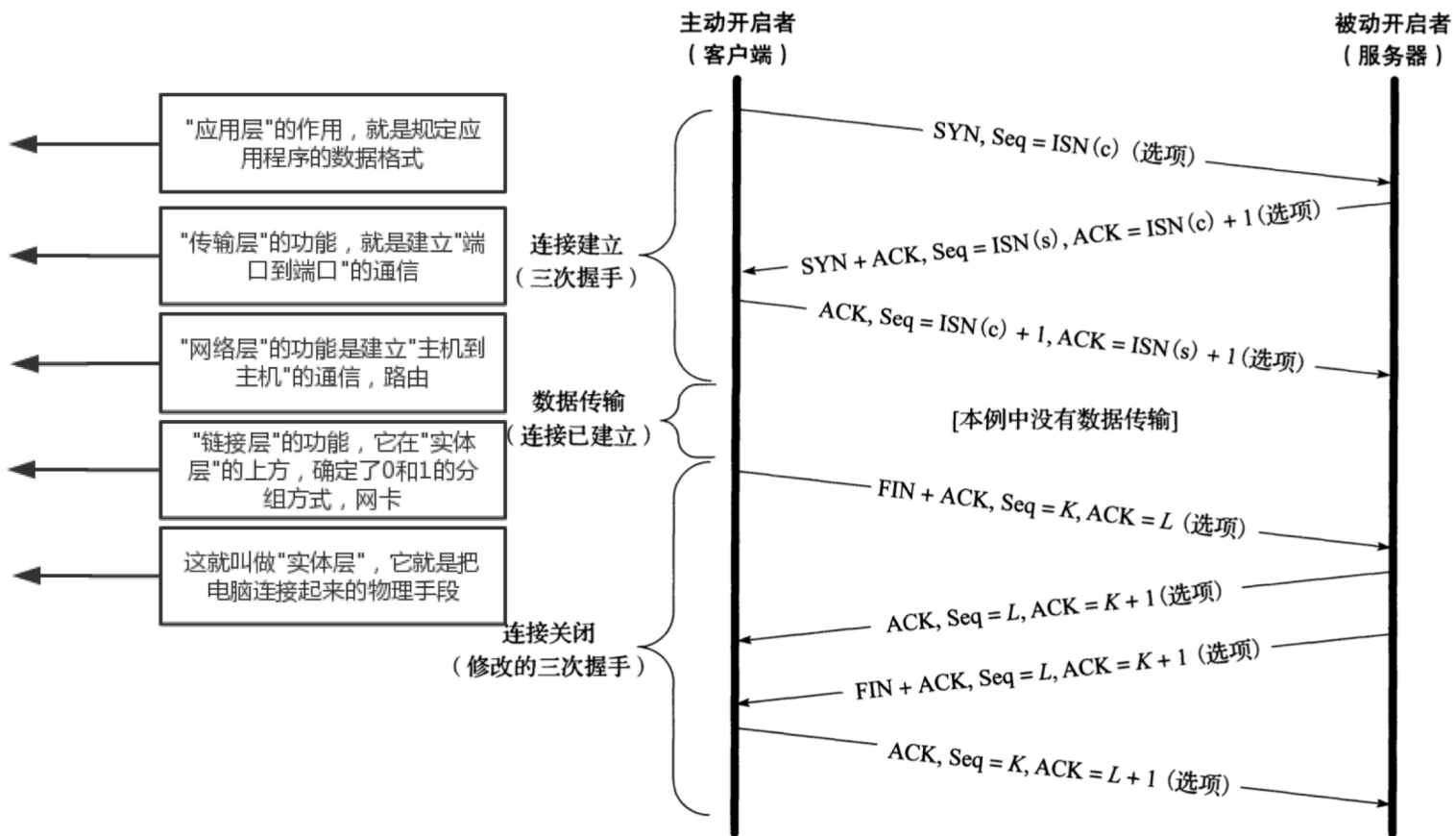
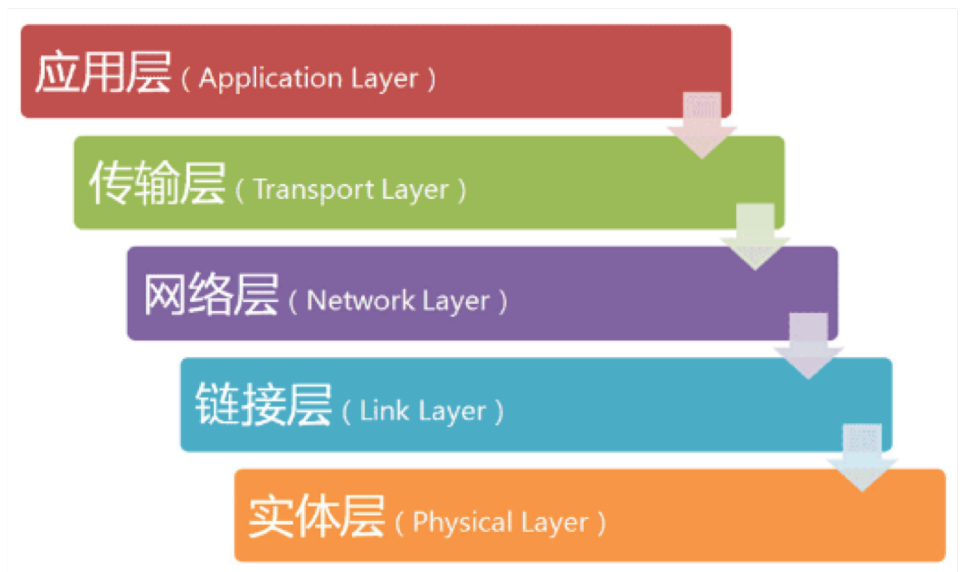
小程序蓝牙开锁 – 协议、拆包、特征值

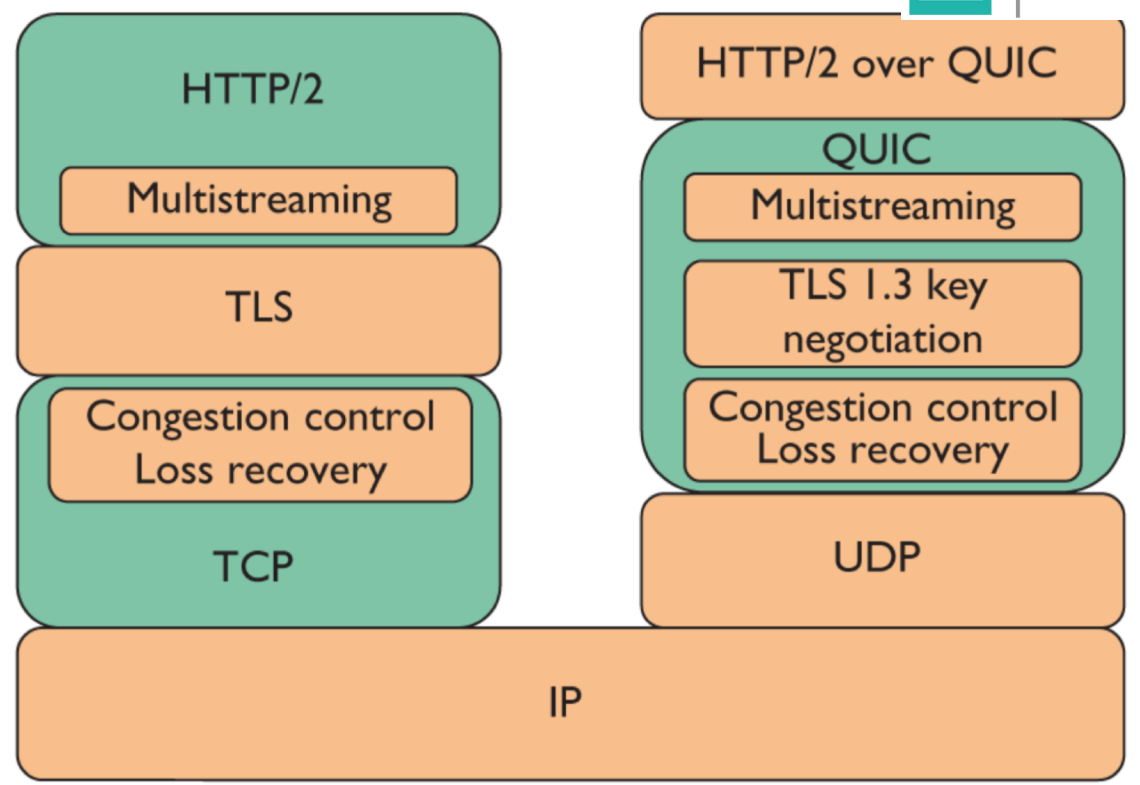
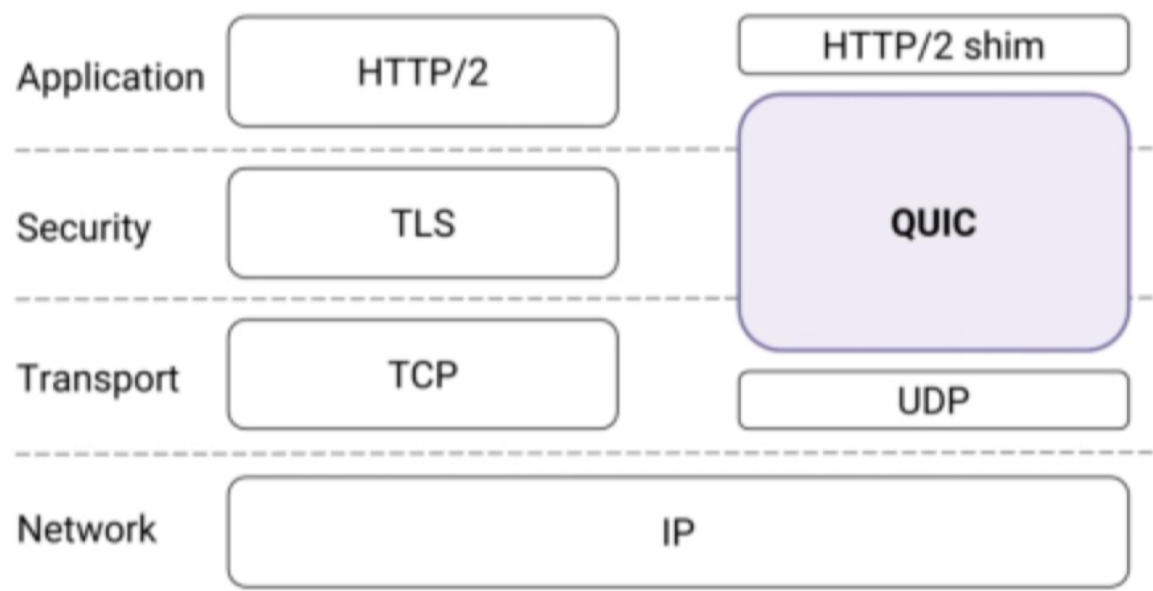
芯片：低功耗 ** 字节

通信与广播，带基础服务，服务端通信加密，海外 dns 解析，proxy 创建到数据收集等

 jweixin-1.3.0.js res.wx.qq.com/open/js	GET	200	spdy	script	index.html?user Parser
 sdk.js [redacted]d...	GET	200 OK	http/1.1	script	index.html?user Parser
 turnplate.aa9749dbd8073f5... 0-[redacted]/active/...	GET	200	h2	stylesh...	index.html?user Parser
 vendor.c86bd6f1a44025ef8... 1-[redacted]active/...	GET	200	h2	script	index.html?user Parser

http 基础





tcp

- ◆ 近三十年发展缓慢
- ◆ 三次握手至少1RTT
- ◆ 窗口慢启动
- ◆ 阻塞拥堵策略无法充分利用带宽

tls

- ◆ tls进一步增加1-2RTT

quic

历史问题

- ◆ 协议历史悠久导致中间设备僵化
- ◆ 依赖于操作系统的实现导致协议本身僵化
- ◆ 建立连接的握手延迟大
- ◆ 队头阻塞

QUIC 协议选择了 UDP，因为 UDP 本身没有连接的概念，不需要三次握手，优化了连接建立的握手延迟，同时在应用程序层面实现了 TCP 的可靠性，TLS 的安全性和 HTTP2 的并发性，只需要用户端和服务端的应用程序支持 QUIC 协议，完全避开了操作系统和中间设备的限制。

quic

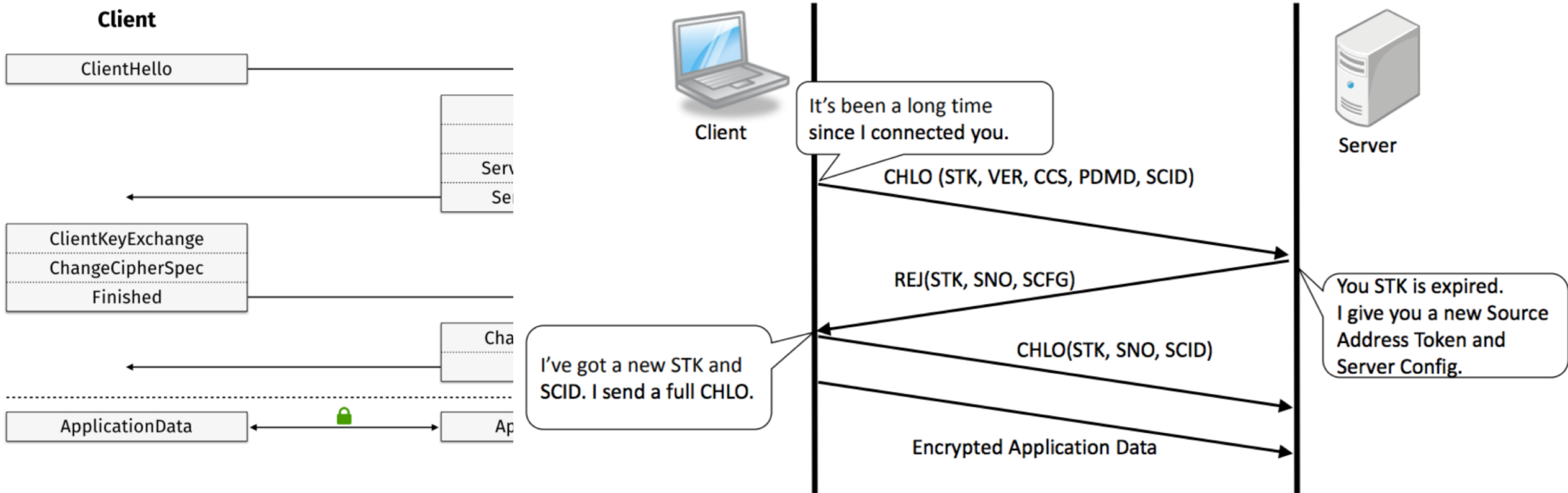
优点

- ◆ 减少了 TCP 三次握手及 TLS 握手时间
- ◆ 改进的拥堵控制
- ◆ 避免队头阻塞的多路复用
- ◆ 连接迁移
- ◆ 向前纠错

缺点

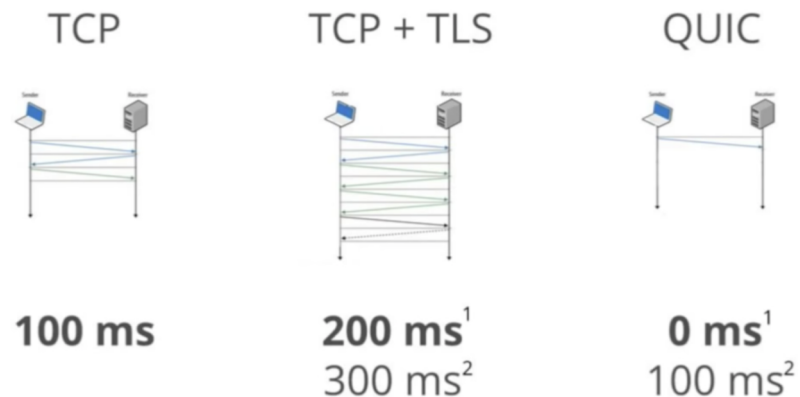
- ◆ 协议分层不清晰，为http/2量身定制
- ◆ UDP流量容易被运营商限制
- ◆ nginx尚无quic支持（腾讯云clb首家支持quic）

quic

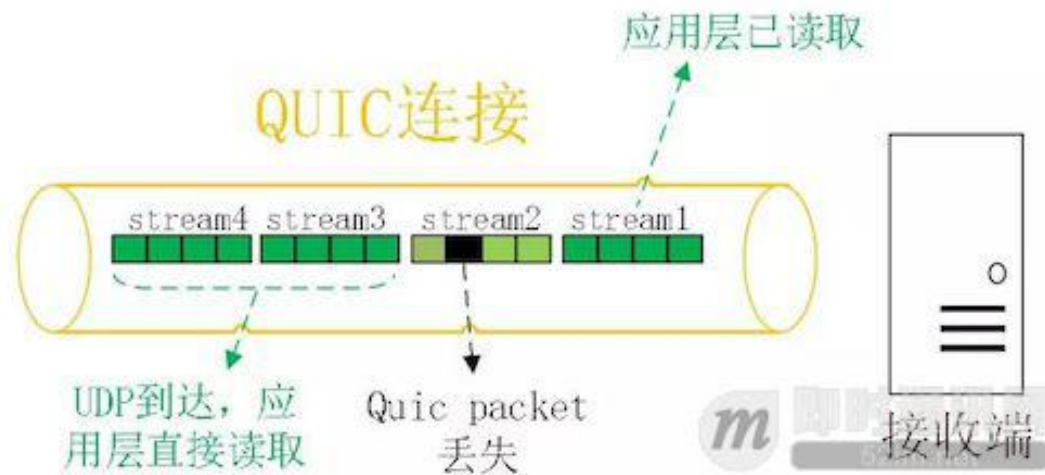
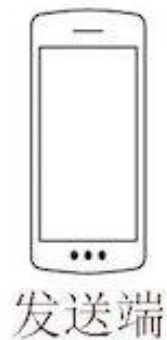
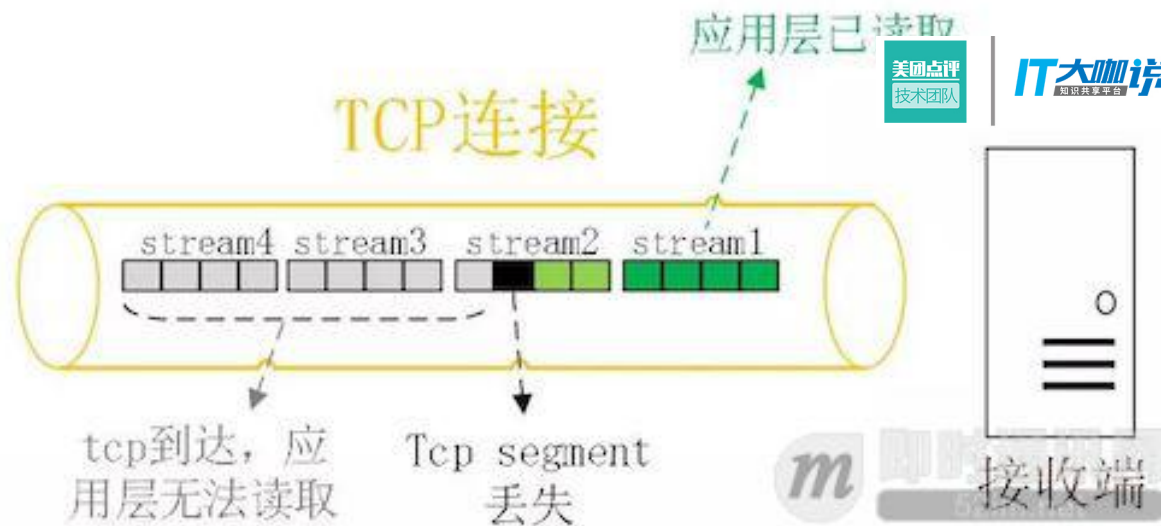
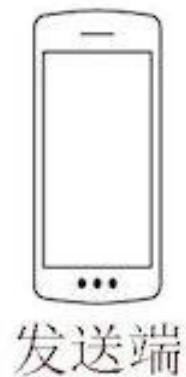


- ◆ HTTPS 的一次完全握手的建连过程，需要 3 个 RTT (tcp 1RTT, tls 2RTT)
- ◆ 而 QUIC 由于建立在 UDP 的基础上只需要1RTT
- ◆ 在恢复会话的情况下，https 需要2RTT，quic 0RTT

quic



各方延迟对比



quic没有队头阻塞的多路复用

多路复用是 HTTP2 最强大的特性
能够将多条请求在一条 TCP 连接上同时发出去。
但也恶化了 TCP 的一个问题，队头阻塞（http2控制stream的粒度与tcp的连接并不搭）

一站式平台化

问题：

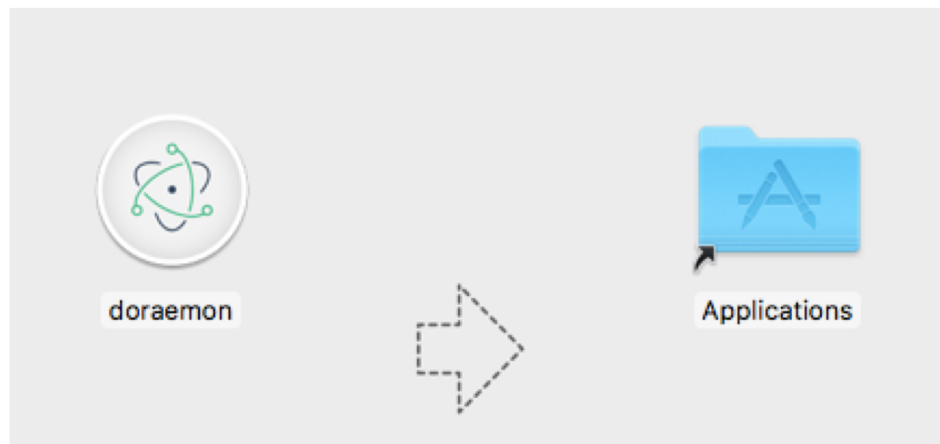
- 1、新员工来了，如何低成本地带进门？
- 2、如何让运营效率得到提升

如何做：

Vue + electron ~

目标：

一个客户端：包含所有~



优化

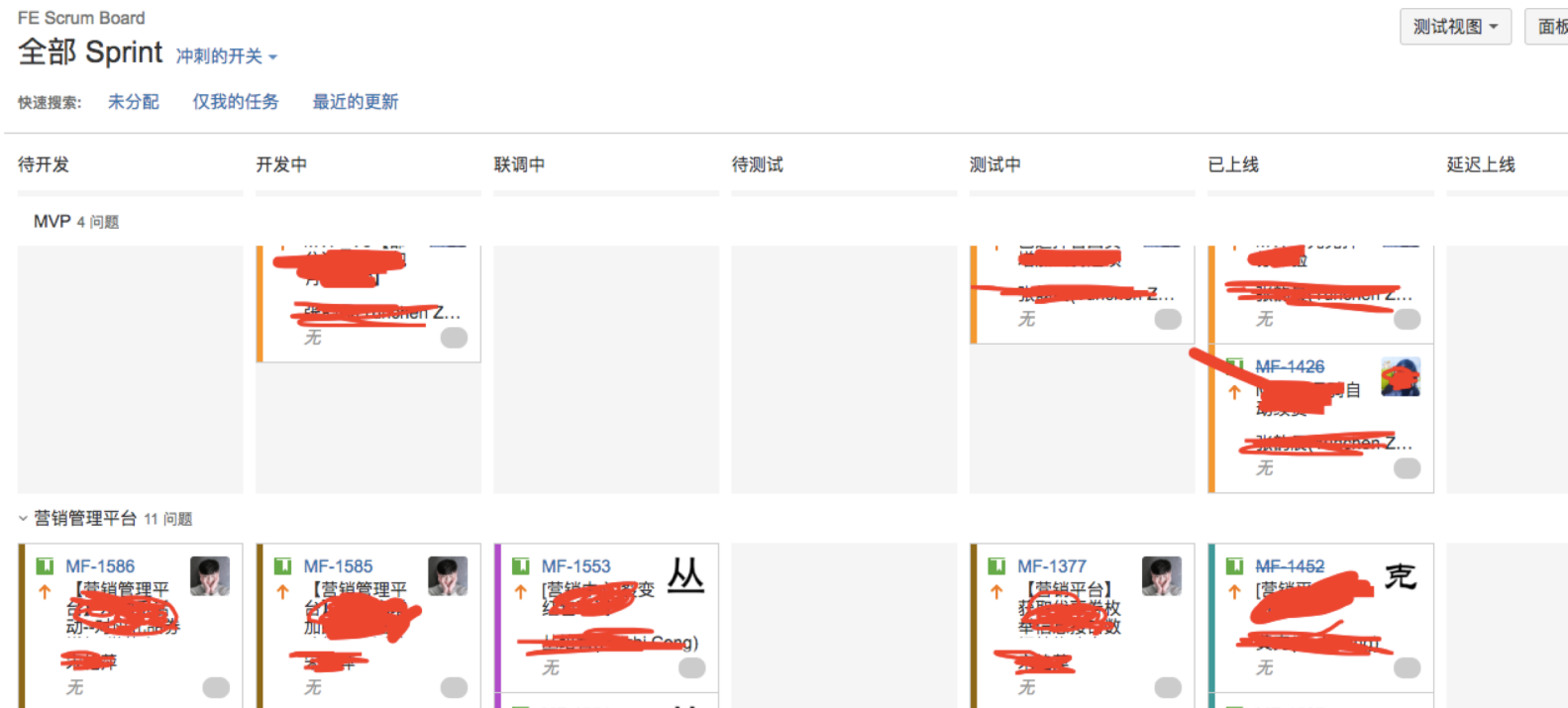
按 service 拆分 main/index.js

```
1 global.services = {  
2   // ...  
3 }
```

更换 renderer ipcRenderer 调用方式

```
1 Vue.prototype.$remoteService =  
2 require('electron').remote.getGlobal('services')
```


敏捷制



前端 RM – release master 制度
人员分布国内国外多有

低频与高频

特种兵与士兵

兴趣与任务

-- 责任和信仰

茶话会

每周1-2次的分享和吐槽，需要看看团队其他人和竞品、行业中在做什么



前端周刊 - 低频

每周一主编，定义文章提交和获奖规则 -- 摩拜前端所有人轮流



前端日刊 - 高频

每天推荐 3 篇文章 -- 摩拜前端日刊小分队（部分人）诚意出品

mobike
摩拜单车

Thanks 大家冒雨而来 & 美团点评邀请

422 摩拜 2 周年，欢迎大家骑摩拜低碳生活