贝壳找房的Node服务稳定性建设

Geekbang. 极客邦科技 InfoQ

极客邦科技
会议推荐2019

5月

QCon 北京
全球软件开发大会
大会：5月6–8日
培训：5月9–10日

QCon 广州
全球软件开发大会
培训：5月25–26日
大会：5月27–28日

6月

GTLC GLOBAL TECH LEADERSHIP CONFERENCE 上海
技术领导力峰会
时间：6月14–15日

GMTC 北京
全球大前端技术大会
大会：6月20–21日
培训：6月22–23日

7月

ArchSummit 深圳
全球架构师峰会
大会：7月12–13日
培训：7月14–15日

10月

QCon 上海
全球软件开发大会
大会：10月17–19日
培训：10月20–21日

11月

GMTC 深圳
全球大前端技术大会
大会：11月8–9日
培训：11月10–11日

AiCon 北京
全球人工智能与机器学习大会
大会：11月21–22日
培训：11月23–24日

12月

ArchSummit 北京
全球架构师峰会
大会：12月6–7日
培训：12月8–9日

自我介绍

徐辛承

- 7年前端工作经验，毕业于北航
- 人人网、百度、百度外卖/饿了么/阿里巴巴，技术专家
- 写过一本书：《Vue移动开发实战技巧》
- 贝壳找房，C端前端负责人

为什么要用Node

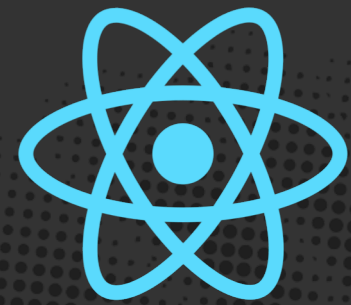
SEO ✓ 服务端渲染

效率 ✓ 前后端同构

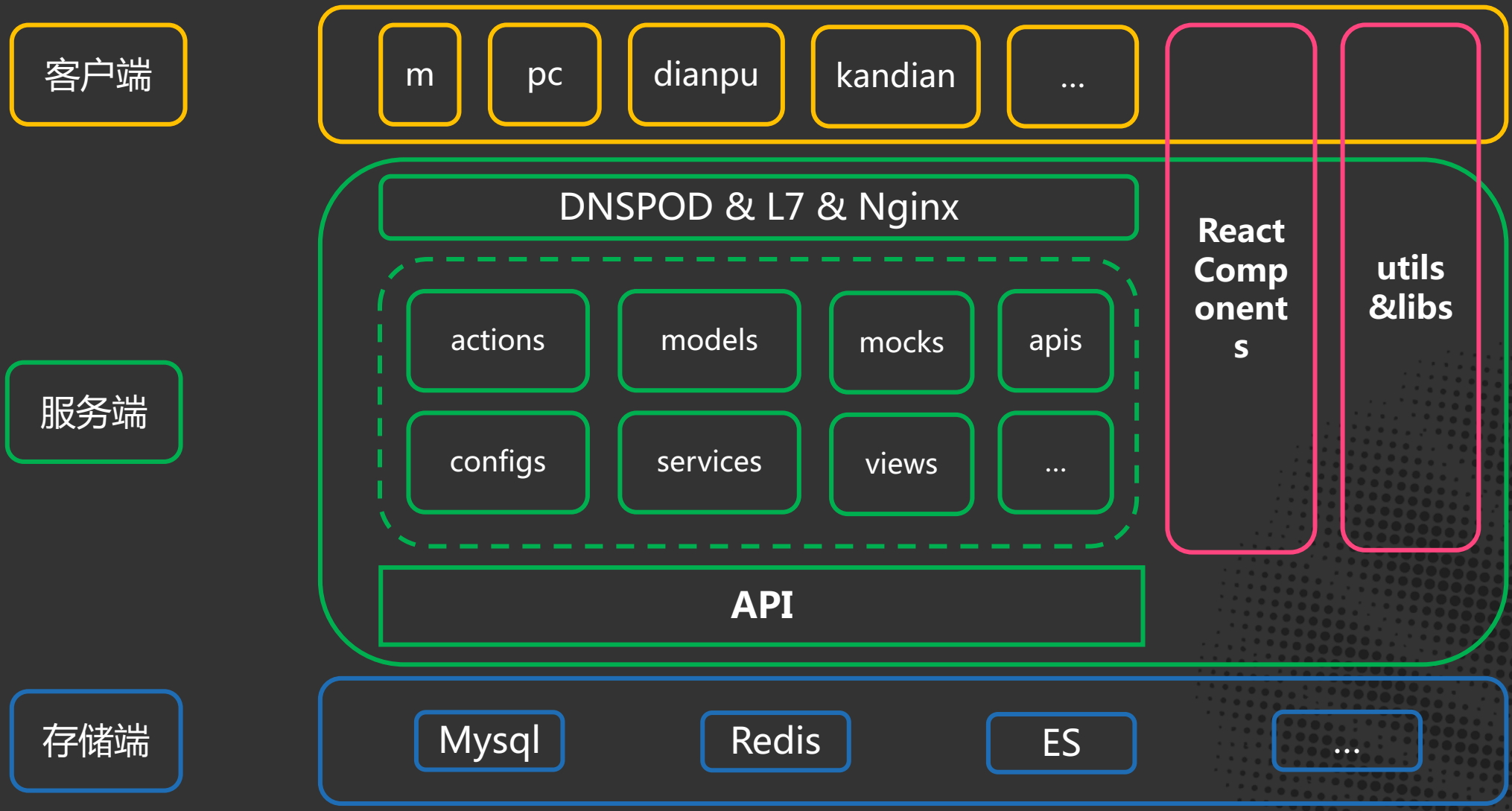
性能 ✓ 首屏更快

Bucky & React

+



技术架构



我们做到了

4000万

月活跃用户

300万

用户push

15万

经纪人整点抢购

稳定性

99.999%

目录

- 预防问题
- 发现问题
- 处理问题
- 总结问题

预防问题

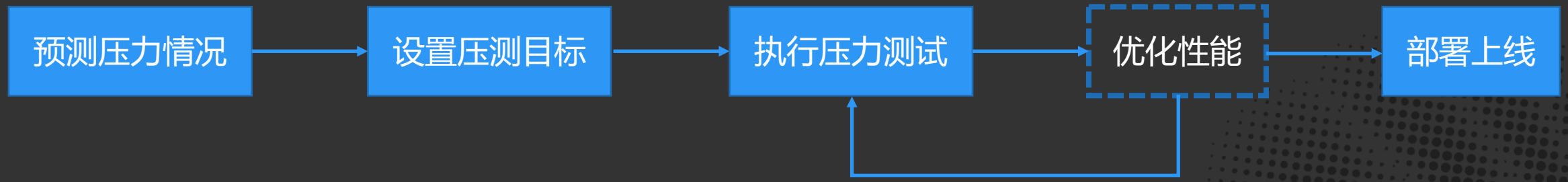
压力估算和压测

Code Review

预防问题：压力估算和压测

- 经纪人店铺新项目上线
- 经纪人抢购

能不能扛得住？



第一步：预测压力情况

- 经纪人店铺
  - 200万请求/天（PV × 平均请求数/页面）
  - 预计80%的流量集中在20%的时间
  - 集群RPS：2000000 × 0.8 ÷ (24 × 0.2 × 3600) ≈ 92.6 RPS
- 经纪人抢购
  - 高峰抢购时大概 2000个用户/s，并发访问量预计翻倍：4000/s
  - 希望返回时间不超过1S
  - 集群RPS：4000 ÷ 1 = 4000 RPS

第二步：设置压测目标

CPU

✓ 多核：<30%

带宽

✓ 不成为瓶颈

内存

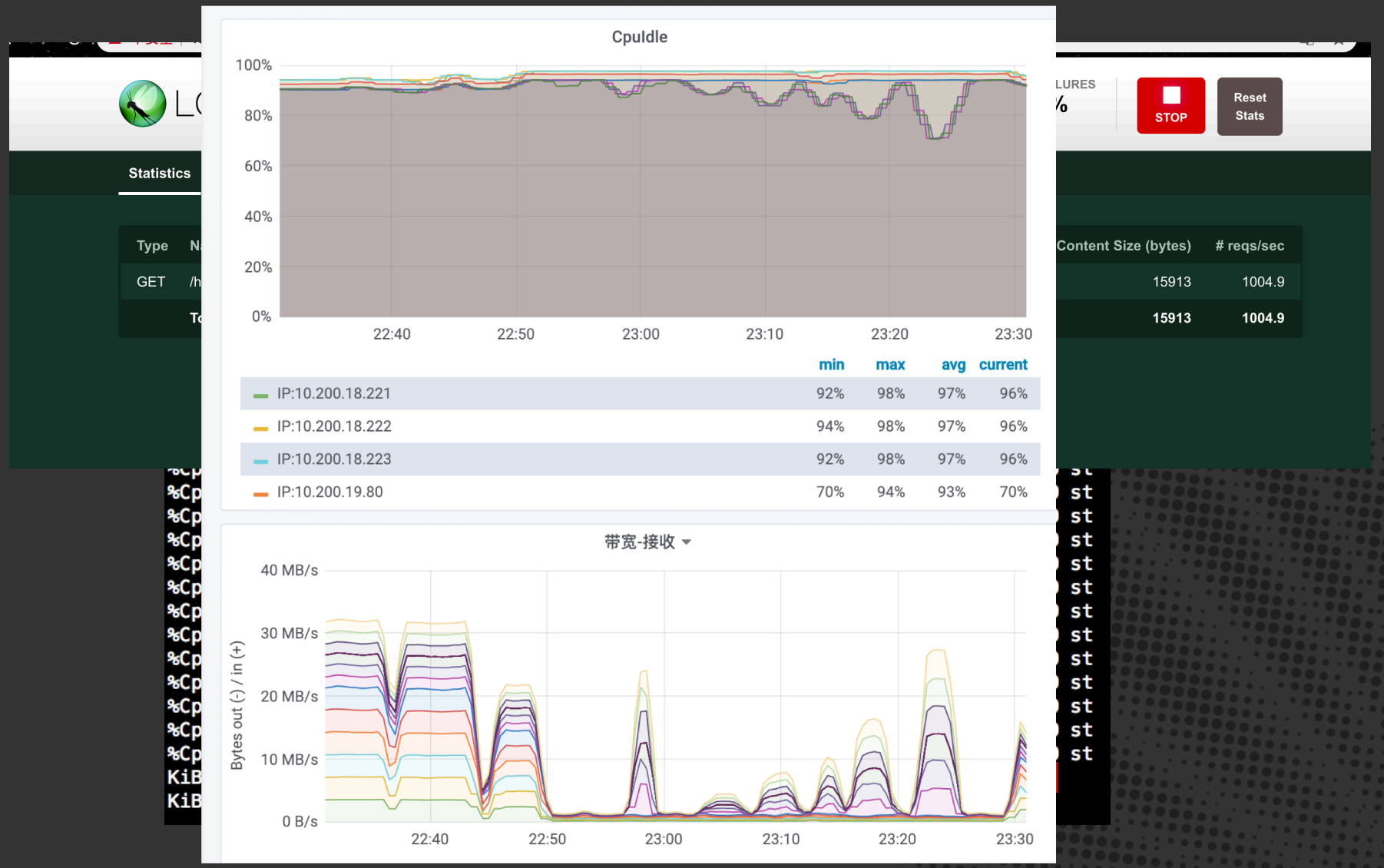
✓ 70%

硬盘

✓ 70%

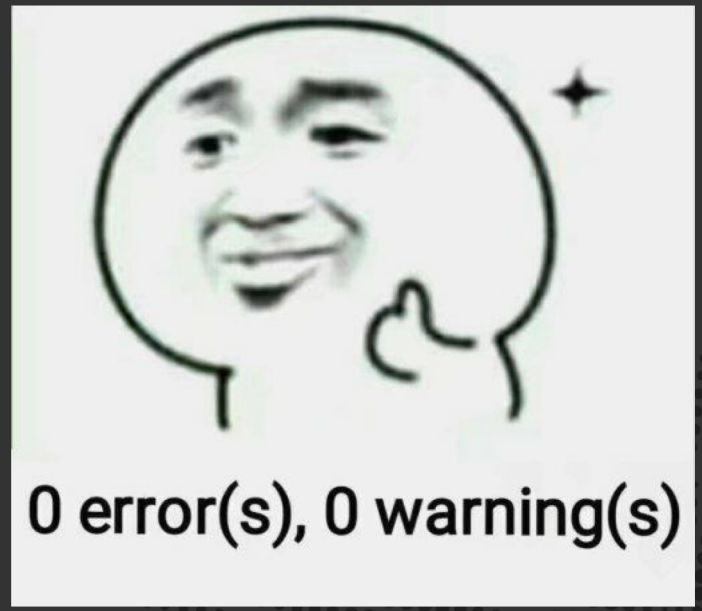
第三步：执行压力测试

- 网络环境相同
- 独立的压力机
- 线上机器
  - ✓ 深夜压测
  - ✓ 混部，周知相关方



第四步：部署上线

- 集群机器数 = 期望RPS ÷ 单机实际RPS ÷ 0.7
- 抢购：集群期望RPS 4000 ÷ 单机实际RPS 800 ÷ 0.7 ≈ 7.14
- 部署7台机器



预防问题：Code Review

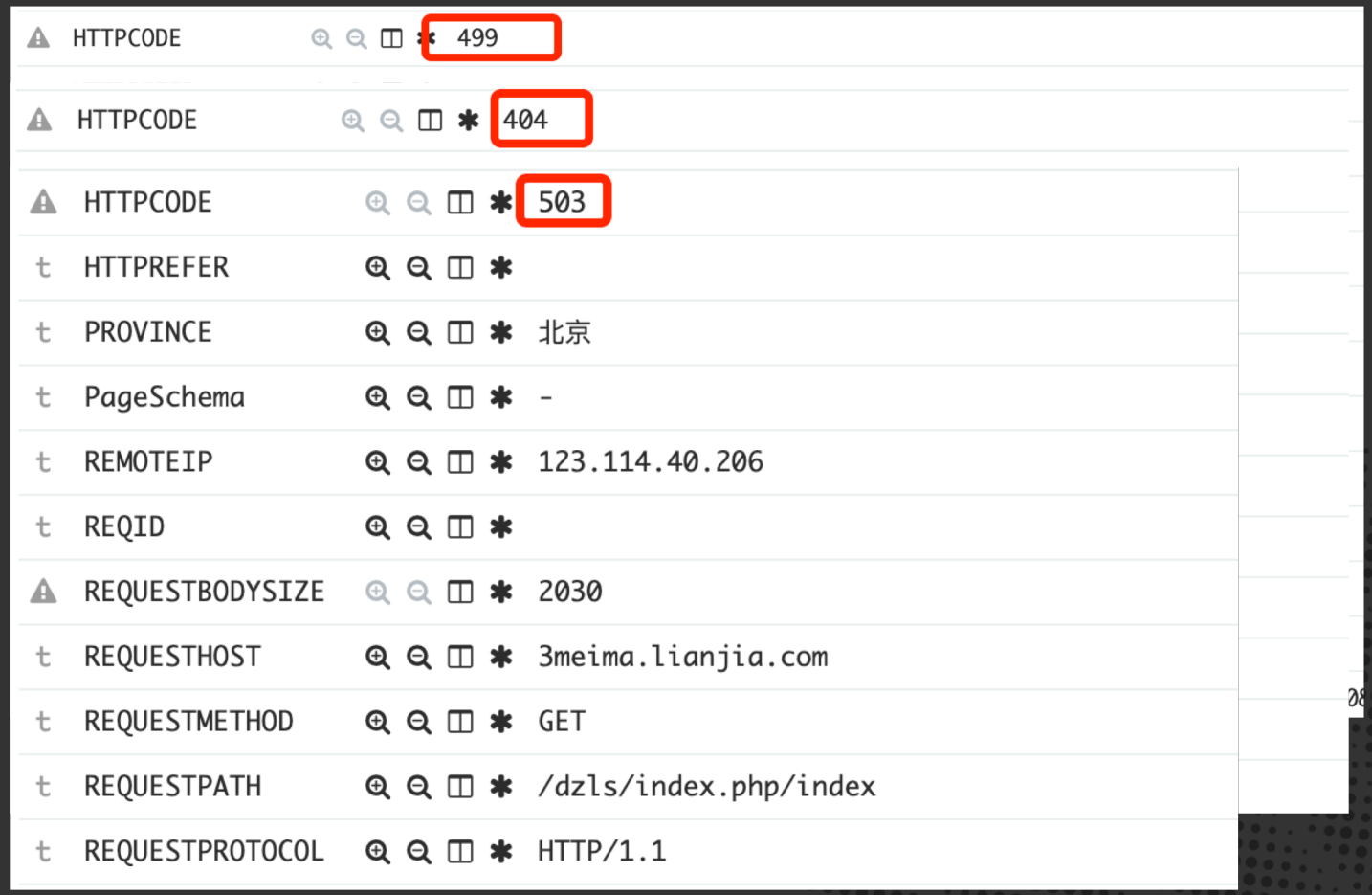
- 变量问题
  - ✓ 根据UA不同，执行不同的逻辑
- 性能问题
  - ✓ async await 串行 || 并行
- 硬编码问题
  - ✓ 环境变量，api domain

```
// 开发环境配置
export const development = api => {
  // api.config.base = 'http://ayx.api.home.ke.com'
  api.config.base = config.devDomain
}

// 测试环境配置
export const testing = api => {
  api.config.base = config.testDomain
}

// 预发环境配置
export const preview = api => {
  api.config.base = 'http://test.api.home.ke.com'
}
```

发现问题

异常监控：监控类型

- Nginx日志
  - ✓ 499
  - ✓ 404
- 服务日志
  - ✓ 5XX



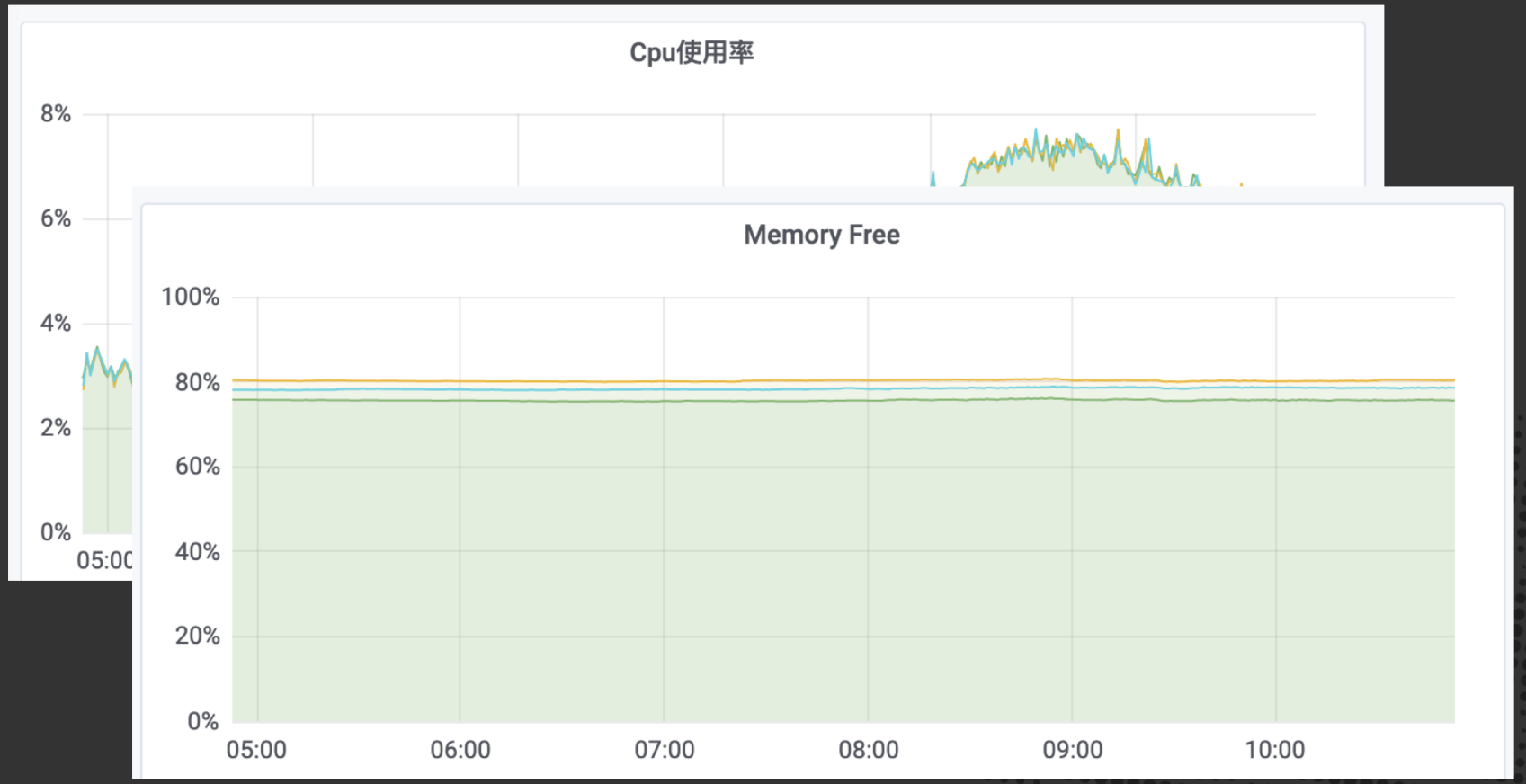
异常监控：Node日志

- Node服务日志
  - ✓ Access.log：客户端每一次请求
  - ✓ Api.log：发起一次api请求
  - ✓ Error.log：所有的错误
- 慎用try catch
  - ✓ Metrics监控


```
"cost": 5011,
"method": "GET",
"actionUri": "http://dianpu.lianjia.com/renderNone",
"ip": "10.26.100.2",
"channel": "request",
"message": "output",
"uri": "http://dianpu.lianjia.com/shop/getbaseinfo/1000001000127830",
"sequence": 5,
"user_id": 0,
"context": "Internal Error<pre>Error: ETIMEDOUT\\n uri:
http://i.purist-api.lianjia.com/seo/pcfooter/innerlink?
cityId=110000&amp;source=ljweb&amp;url=http%3A%2F%2Fbj.lianjia.com&amp;
request_source=ljweb\\n displayName: API.ljFooter.getPCFooter\\n file:
/data0/www/htdocs/dianpu.ke.com/dist/server/apis/ljFooter.js\\n\\n\\n
${app}/node_modules/request/request.js:849:19\\n\\n 845 | // time, not
the time between bytes sent by the server.\\n 846 | self.timeoutTimer =
setTimeout(function () {\\n 847 |
socket.removeListener(&#039;connect&#039;, onReqSockConnect)\\n 848 |
self.abort()\\n&gt; 849 | var e = new Error(&#039;ETIMEDOUT&#039;)\\n |
^\\n 850 | e.code = &#039;ETIMEDOUT&#039;\\n 851 | e.connect = true\\n
852 | self.emit(&#039;error&#039;, e)\\n 853 | }, timeout)\\n 854 | }
else {\\n\\n at Timeout._onTimeout
(${app}/node_modules/request/request.js:849:19)\\n at ontimeout
(timers.js:469:11)\\n at tryOnTimeout (timers.js:304:5)\\n at
Timer.listOnTimeout (timers.js:264:5)</pre>",
"request_id": "c3e0bc52-8356-45a6-b616-adaf1158c0b7",
"timestamp": "2019-06-03T17:06:13.000+08:00"
```

发现问题：资源监控

- CPU
  - ✓ 40%
- Memory
  - ✓ 70%

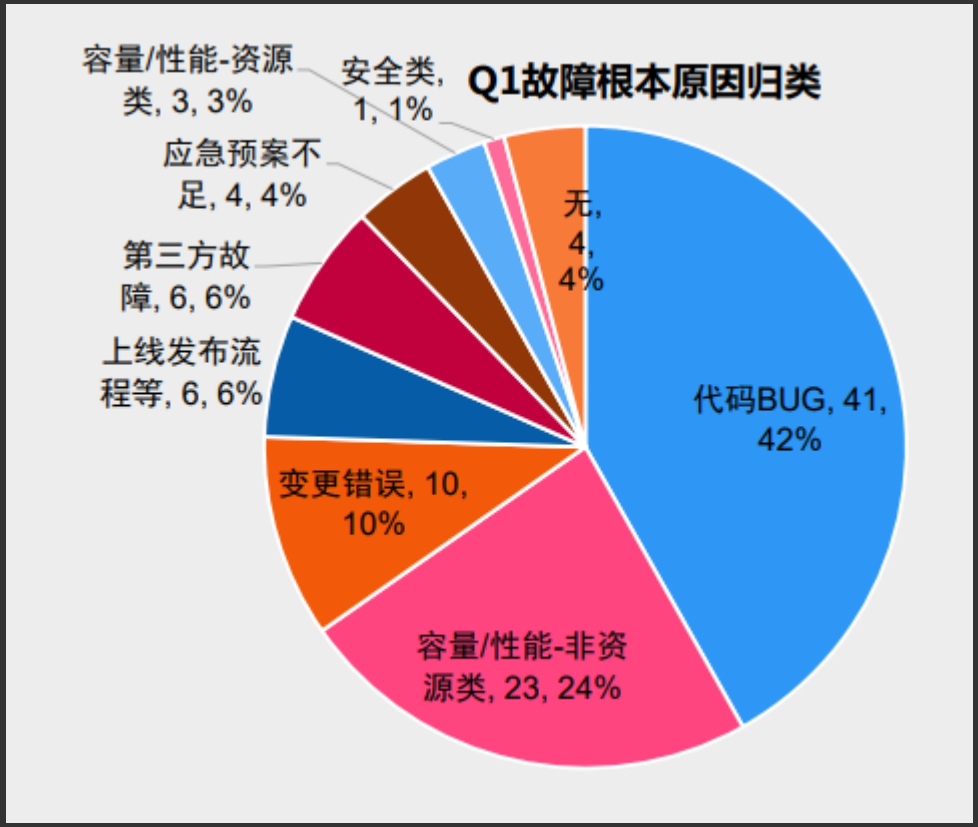


处理问题

原则：恢复大于原因查找

- 快速回滚
- 快速定位
- 重启和扩容

处理问题：快速回滚



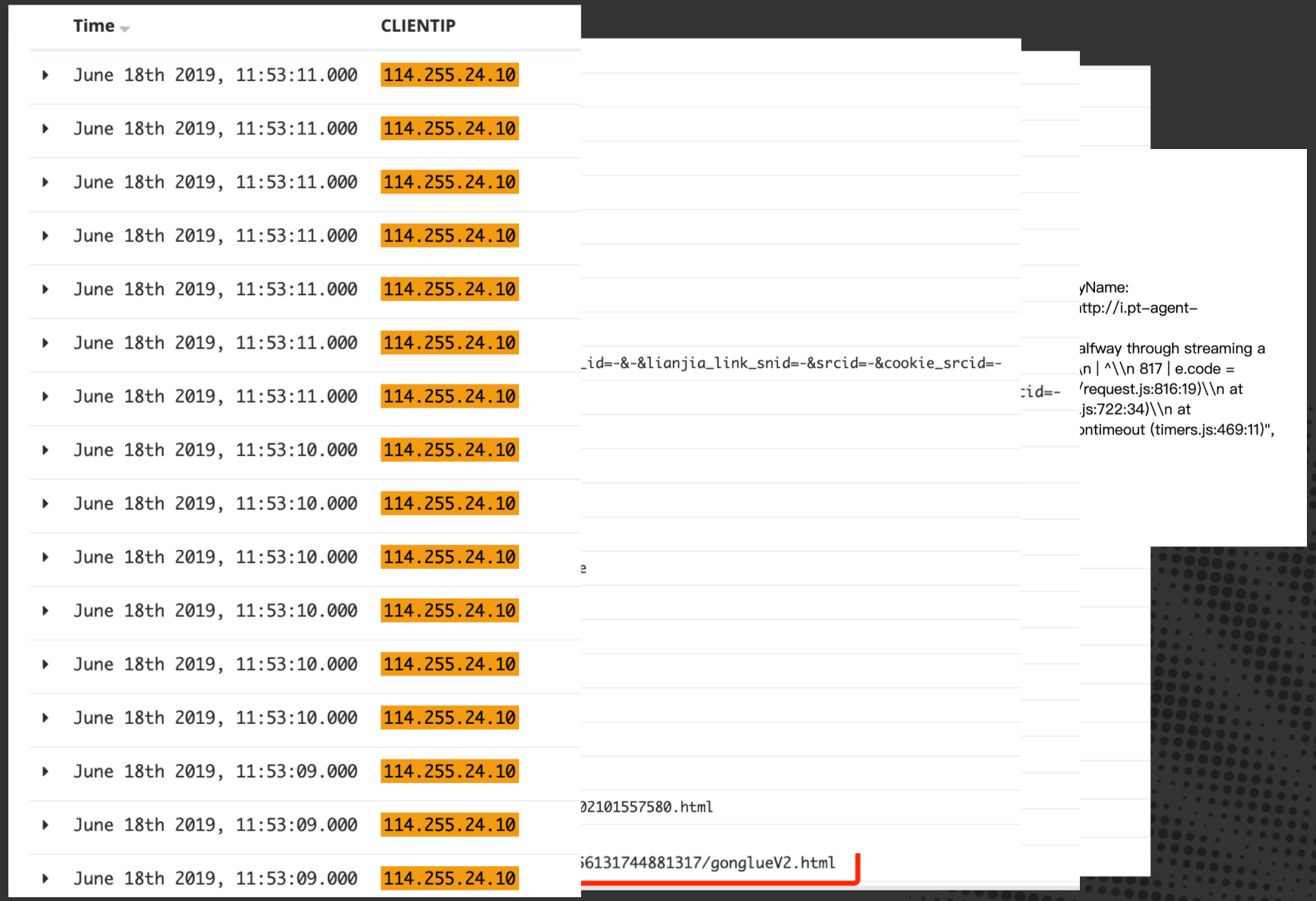
快速回滚

发布平台

回滚步骤

快速定位：日志检索

- 错误信息和请求信息
- 499
  - 主动断开
  - 大量超时
- 5XX
- 404
  - 上线引发的路由问题
  - 大量爬虫

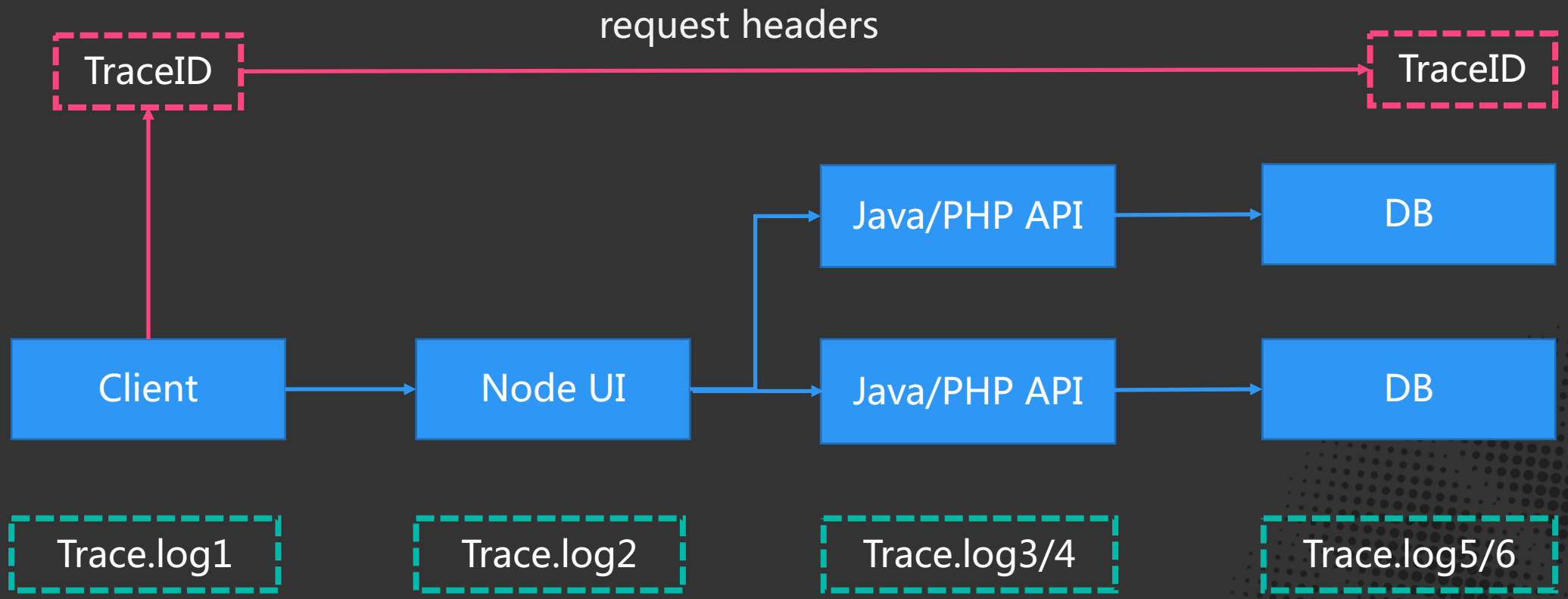


快速定位：无法直观看出的问题

t	_id	AWtK-7vYNgjrvEijE_Kh
t	_index	index-719-967_2019-06-12
#	_score	-
t	_type	index-719-967
t	channel	error
t	context	Cannot read property '0' of undefined\nTypeError: Cannot read property '0' of undefined\n\n ${app}/dist/server/models/DesignerCase.js:98:37\n\n 94 | // k.url = k.url.split('!')[0] + '!m_fill,w_' + bigImgWidth\n 95 | });\n 96 | } else if (v.type === 'furniture') {\n 97 | v.list && v.list.forEach(function (m) {\n> 98 | if (m.url_list[0].url.indexOf('.gif') < 0) {\n | ^\n 99 | var resp2 = (0, _utils.dealPic)(m.url_list[0].url.split('!')[0], 192, 144, dpr);\n 100 | m.url_list[0].url = resp2.url;\n 101 | m.url_list[0].cutFlag = resp2.cutFlag;\n 102 | m.url_list[0].cutItem = resp2.cutItem;\n 103 | }\n\n at ${app}/dist/server/models/DesignerCase.js:98:37\n at Array.forEach (<anonymous>)\n at ${app}/dist/server/models/DesignerCase.js:97:48\n at Array.forEach (<anonymous>)\n at ${app}/dist/server/models/DesignerCase.js:86:54\n at Array.forEach (<anonymous>)\n at ${app}/dist/server/models/DesignerCase.js:84:48\n at Array.forEach (<anonymous>)\n at Function._callee$ (${app}/dist/server/models/DesignerCase.js:68:24)\n at tryCatch (${app}/node_modules/regenerator-runtime/runtime.js:62:40)
t	message	TypeError
t	request_id	c4c1120b-3ed0-4a63-9a1d-a070a563ea17
t	segmentId	home.newm.ke.com-10.200.18.221-55033-1560331136910-Gp0Ov8aZiA
t	spanId	0
🕘	timestamp	June 12th 2019, 17:18:56.000
t	traceId	home.newm.ke.com-10.200.18.221-55033-1560331136910-3RvWPw6Afp
t	uri	http://home.newm.ke.com/page/designer-case

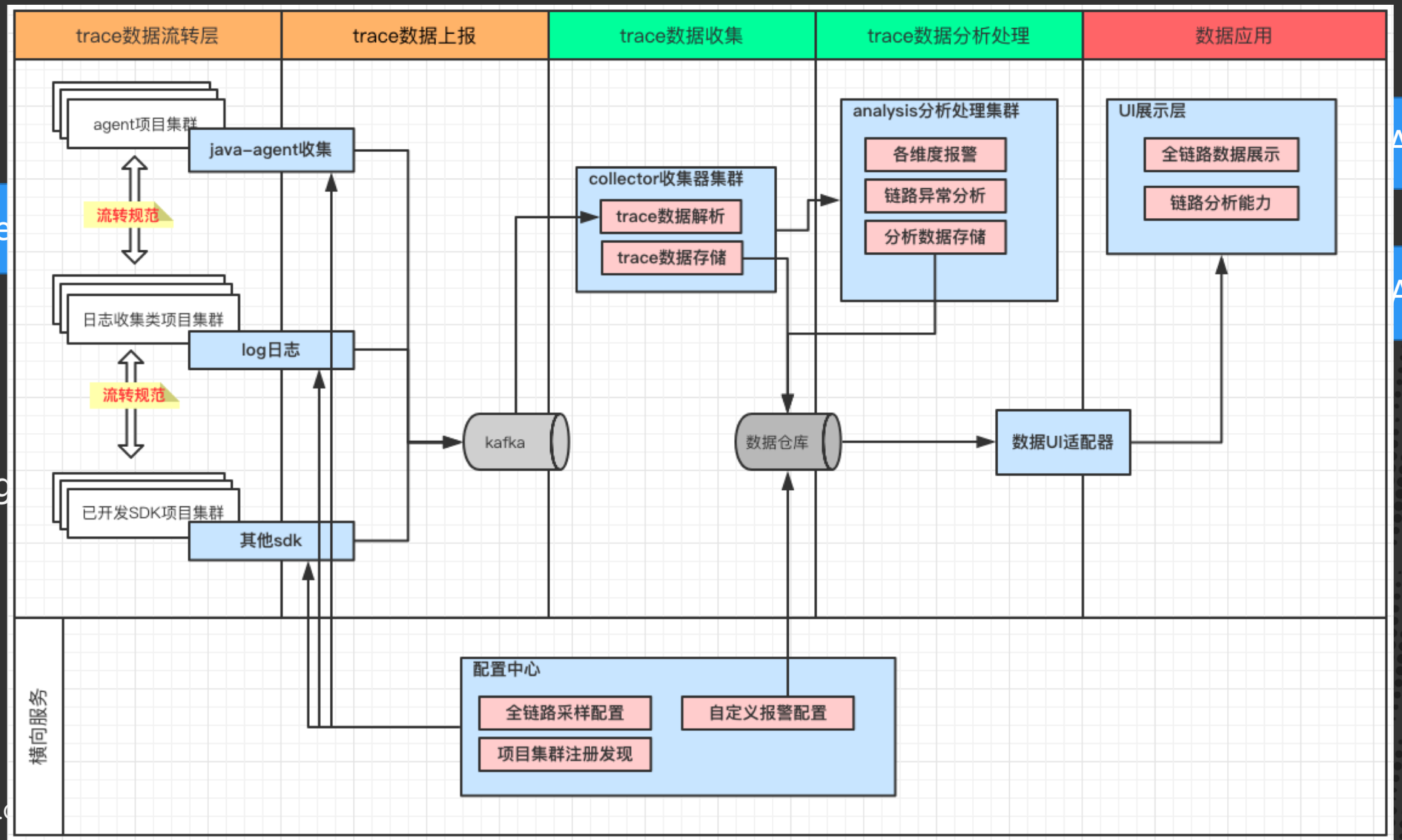
- ✓ 看代码，可能哪个变量有问题
- ✓ 根据请求链路向下追溯

快速定位：链路追踪

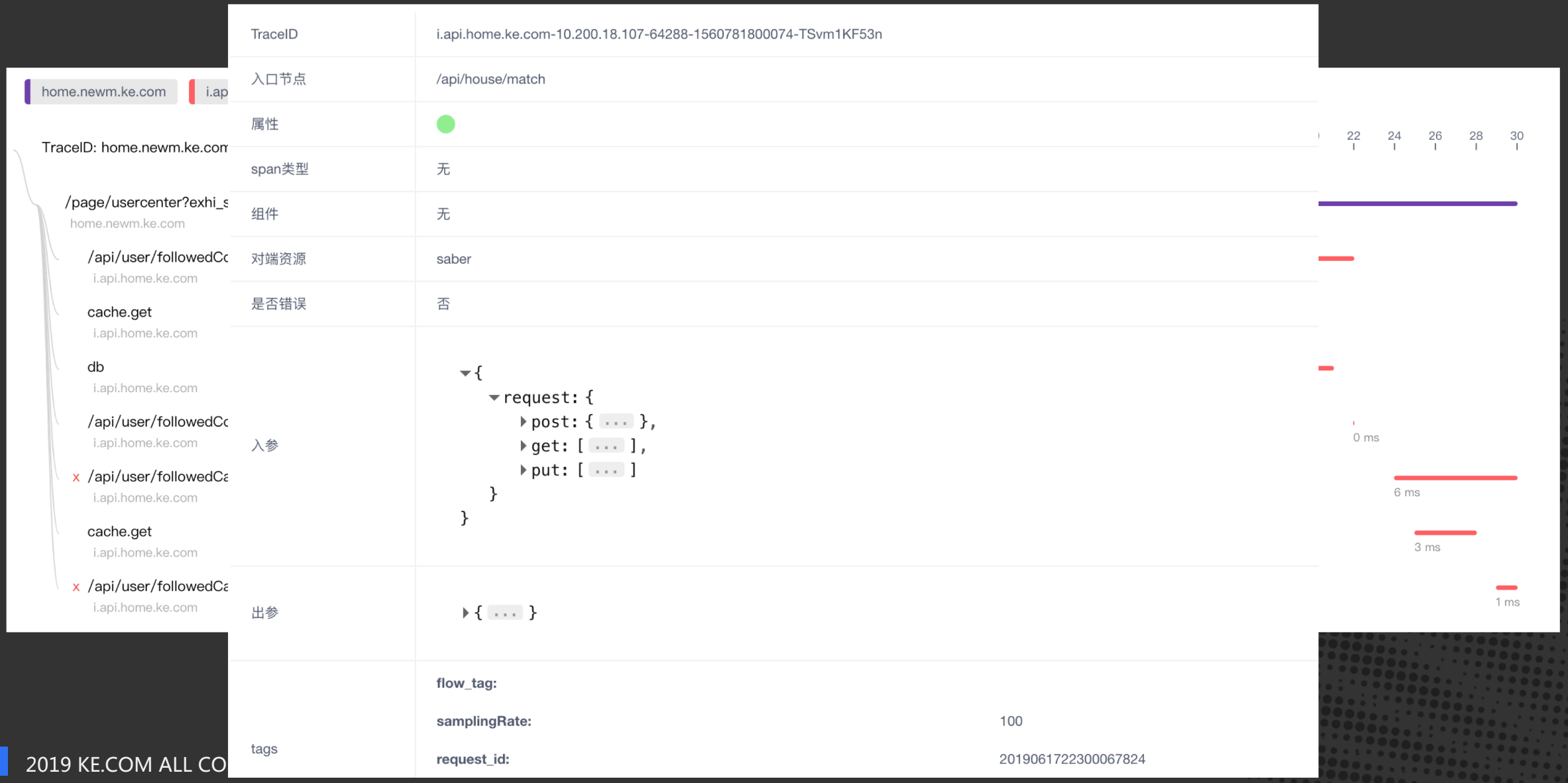


- ✓ TraceID作为唯一标示串起整个链路
- ✓ 每个节点详细信息以log形式输出

快速定位：Ketrace



快速定位：链路追踪



处理问题：重启和扩容

内存占用过高

- 是否内存泄漏
- 重启服务

CPU Idle 过低

- 快速扩容
- 周期性部署
  - ✓ 预部署 + 周期性切流

总结问题

Case Study

- ✓ 处理过程：时间、人员、方式
- ✓ 直接原因和根本原因
- ✓ 可执行的整改计划

分享总结

- 预防问题
  - ✓ 压力预估和压力测试
  - ✓ Code Review
- 发现问题
  - ✓ 异常监控
  - ✓ 资源监控
- 处理问题
  - ✓ 快速回滚
  - ✓ 快速定位
  - ✓ 重启和扩容
- 总结问题
  - ✓ Case Study

展望

- 预防问题
  - ✓ 压测平台
  - ✓ 仿真环境&故障预演
  - ✓ 自动化测试
- 发现问题
  - ✓ 服务端埋点
  - ✓ 线上巡检
- 处理问题
  - ✓ 降级、限流、熔断、扩容、静态化预案
  - ✓ 配置中心&预案平台


贝壳
找房大平台