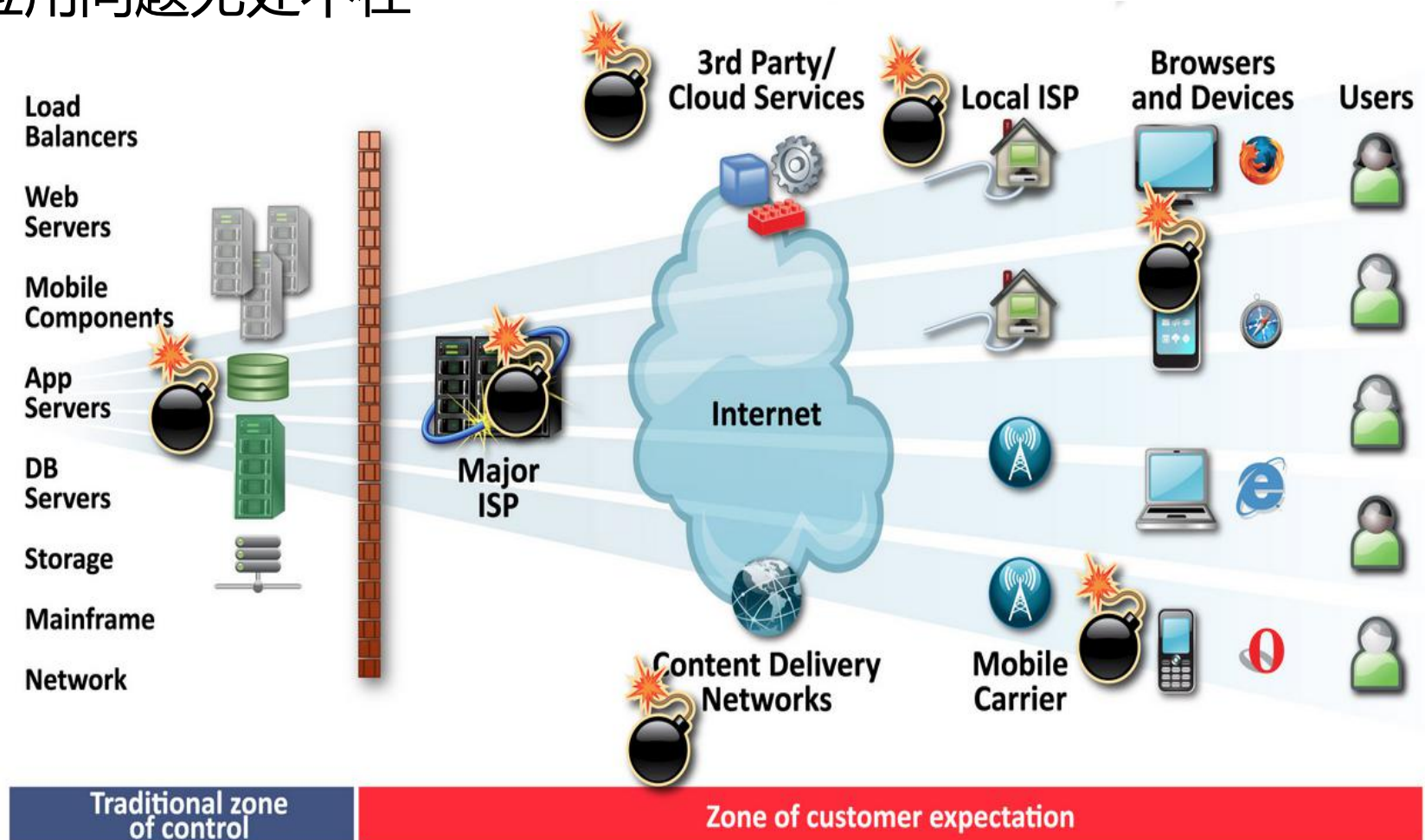


# APM全栈性能监控实践

廖雄杰@听云  
研发副总裁

## 应用问题无处不在

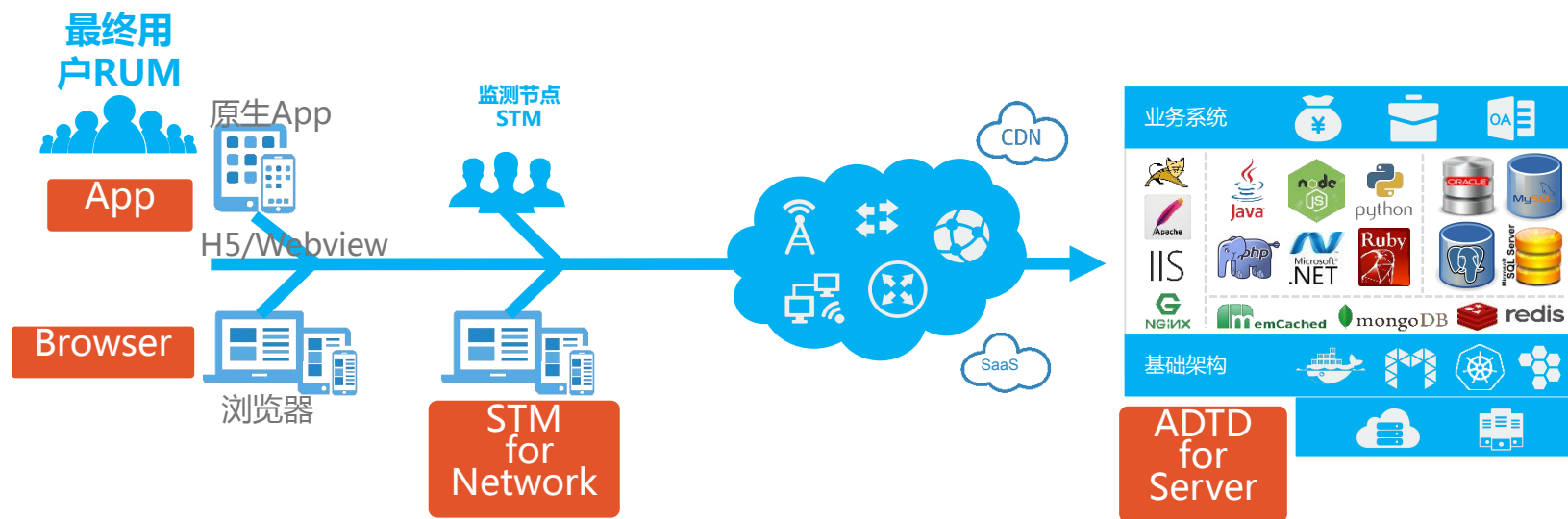


## 如何监控发现应用性能及问题

- 数字化体验监控 ( Digital experience monitoring , DEM)
  - 真实用户监控 ( Real User Monitoring, RUM )
  - 模拟事务监控 ( Synthetic Transaction Monitoring, STM )
- 应用发现、跟踪和诊断 ( Application discovery, tracing and diagnostics, ADTD)
- 应用分析 ( Application analytics , AA)

\* 摘自Gartner对APM套件核心功能的定义

## APM全栈溯源的几个步骤



## APM全栈溯源的几个步骤

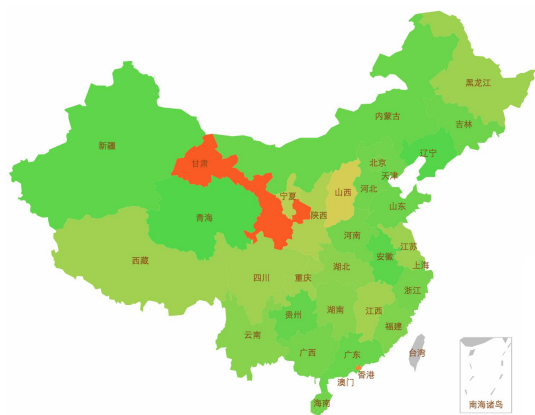
1. 用户体验一览

2. 真实用户体验

3. 网络切片

4. 应用拓扑

5. 行级代码分析



85.33ms



响应时间

250%



崩溃率

活跃版本数: 1

首次接收数据: 2016-03-07

0%



HTTP错误率

0%



网络错误率

实例数量: 1

吞吐率: 28.93

928ms



响应时间

0.69



Apdex

0%



错误率

## APM全栈溯源的几个步骤

1. 用户体验一览    2. 真实用户体验    3. 网络切片    4. 应用拓扑    5. 行级代码分析

### 交互性能分析-IOS

孕育管家 iPhone

排序 执行时间 隐藏<1%

视图一览

PHKnowledge...r#AfterLoading	2.01秒
PHKnowledge...troller#loading	0.885秒
PHForumPost...troller#loading	0.756秒
PHAdSplashV...troller#loading	0.732秒
PHTabbarView...troller#loading	0.642秒
PHPHForumsC...SXVC#loading	0.624秒
PHForumRepl...troller#loading	0.605秒
PHCollectionV...troller#loading	0.559秒
PHNewForumsListVC#loading	0.429秒
PHForumHotS...troller#loading	0.399秒
PHForumHom...troller#loading	0.379秒
PHEatListView...troller#loading	0.365秒
PHForumMine...troller#loading	0.299秒
PHForumInter...troller#loading	0.149秒

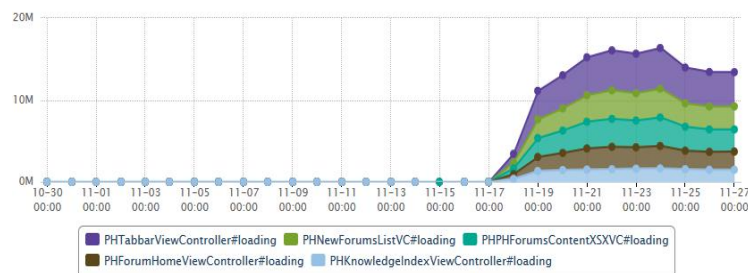
### 页面加载分析-IOS

排序 页面加载 隐藏<1%的吞吐量

性能一览

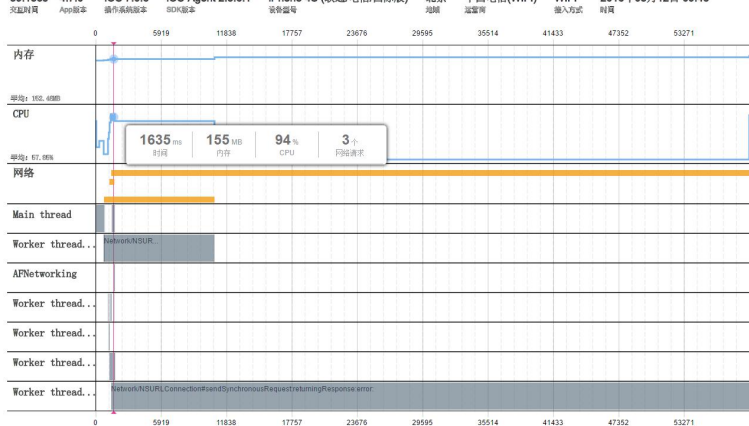
全部	自身服务	第三方
act.xiaoshuxiong.com 3.725秒		
来源：自身服务		
m.mama.cn 2.497秒		
来源：自身服务		
www.xiaoshuxiong.com 1.581秒		
来源：自身服务		
papi.mama.cn 1.178秒		
来源：自身服务		
app.mama.cn 0.797秒		
来源：自身服务		
www.mama.cn 0.608秒		
来源：自身服务		

TOP5 最热门视图



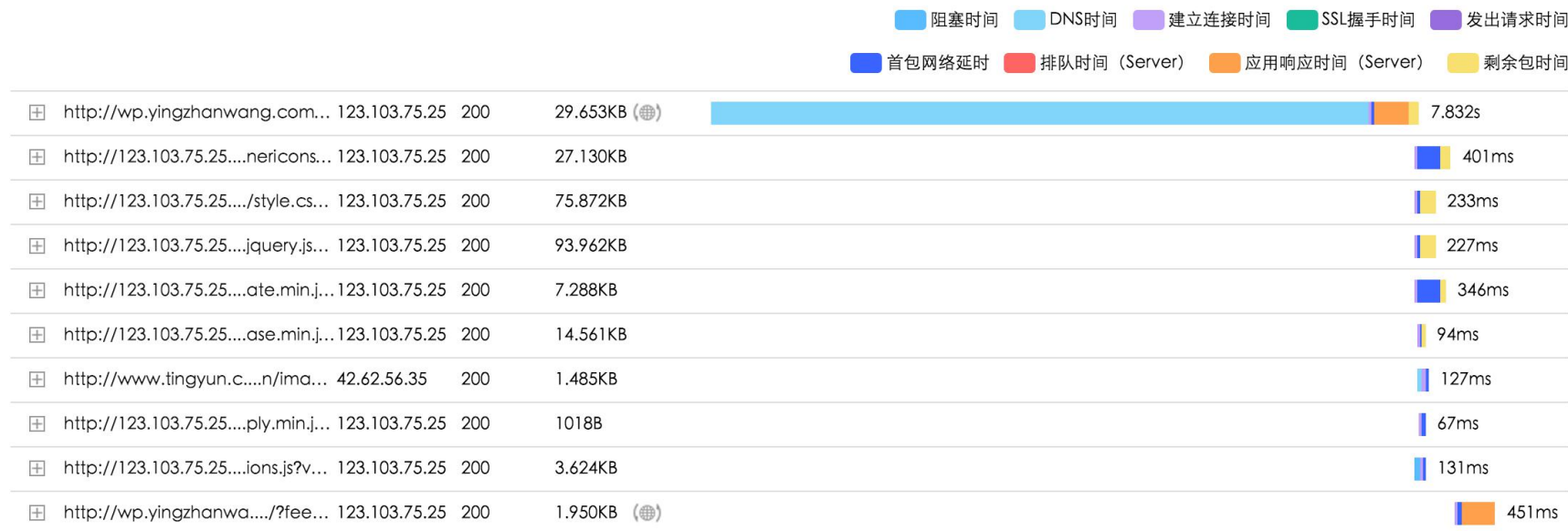
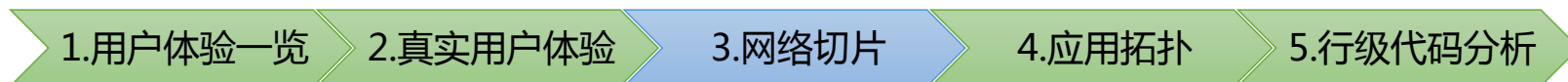
MyTabBarViewController#loading

59.188s 4.7.0 iOS 7.0.6 iOS-Agent 2.3.6.1 iPhone 4S (联通/电信/国际版) 北京 中国电信(WIFI) WIFI 2016年08月12日 09:43





## APM全栈溯源的几个步骤



## APM全栈溯源的几个步骤

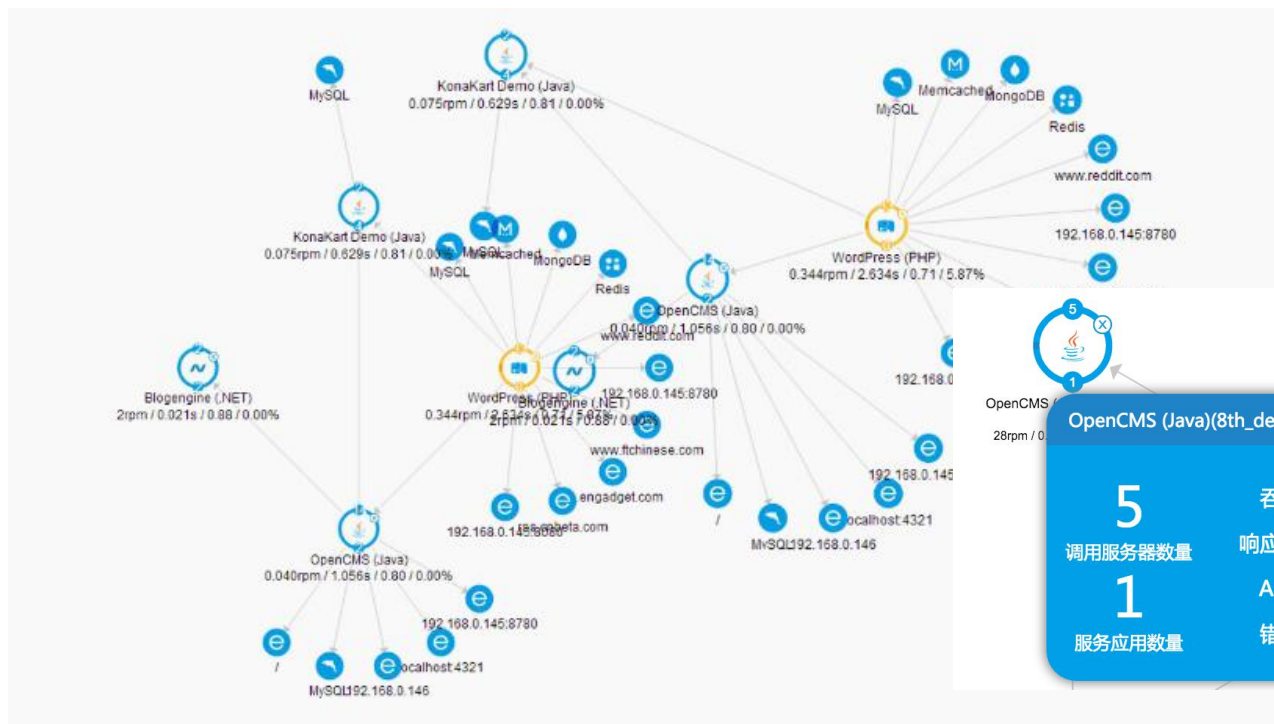
1. 用户体验一览

2. 真实用户体验

3. 网络切片

4. 应用拓扑

5. 行级代码分析



OpenCMS (Java)(8th\_demo) [查看>](#)

5	吞吐量	28rpm
调用服务器数量	响应时间	0.755s
1	Apdex	0.83T:1000ms
服务应用数量	错误率	2.91%



## APM全栈溯源的几个步骤

1. 用户体验一览

2. 真实用户体验

3. 网络切片

4. 应用拓扑

5. 行级代码分析

▼ get_template_part	≡	2501	51.42	2305
▼ locate_template	≡	2501	51.42	2305
▼ load_template	≡	2501	51.42	2305
▼ require	≡	2501	51.42	2305
▼ the_content	≡	2492	51.23	2313
▼ apply_filters	≡	2490	51.19	2313
▼ call_user_func_array	≡	2459	50.56	2313
▼ ExecPhp_Runtime.filter_user_content	≡	2459	50.56	2313
▼ ExecPhp_Runtime.eval_php	≡	2458	50.53	2314
▼ eval	≡	2457	50.51	2314
file_get_contents	≡	2457	50.51	2314

外部应用: [OpenCMS \(Java\)](#) 实例信息: fe80:0:0:0:250:56ff:fe97:70c6%2:8080

Web应用过程: [WebAction/JSP/WEB-INF/jsp/online/sites/default/\\_konakart\\_002dintegration/index.jsp](#)

URL: <http://192.168.0.145:8080/opencms/opencms/konakart-integration/>



数据库调用时间 应用层时间 外部服务时间

## APM全栈溯源的几个步骤

- 真实用户性能：DEM/RUM
- 网络切片：STM/NPM
- 后台应用逻辑拓扑：ADTD
- 应用过程及代码级分析：ADTD

# 浏览器->服务器溯源

## 浏览器->服务器溯源

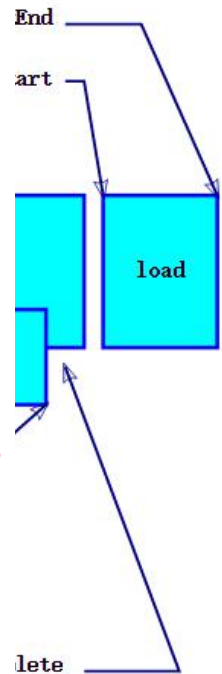
```
> window.performance.getEntriesByType("resource")
(48) [PerformanceResourceTiming, PerformanceResourceTiming, PerformanceResourceTiming, ..., l]
  ▼ 0: PerformanceResourceTiming {initiatorType: "link", workerStart: 0, redirectStart: 0, ...}
  ▼ 1: PerformanceResourceTiming {initiatorType: "link", workerStart: 0, redirectStart: 0, ...}
  ▼ 2: PerformanceResourceTiming {initiatorType: "link", workerStart: 0, redirectStart: 0, ...}
  ▼ 3: PerformanceResourceTiming {initiatorType: "img", workerStart: 0, redirectStart: 0, ...}
  ▼ 4: PerformanceResourceTiming
    connectEnd: 1415.41
    connectStart: 1415.41
    decodedBodySize: 10755
    domainLookupEnd: 1415.41
    domainLookupStart: 1415.41
    duration: 844.24
    encodedBodySize: 10755
    entryType: "resource"
    fetchStart: 1415.41
    initiatorType: "img"
    name: "http://2017.thegiac.com/images/top_giac.png"
    redirectEnd: 0
    redirectStart: 0
    requestStart: 1467.135
    responseEnd: 2259.65
    responseStart: 1467.8000000000002
    secureConnectionStart: 0
    startTime: 1415.41
    transferSize: 0
    workerStart: 0
    __proto__: PerformanceResourceTiming
  ▼ 5: PerformanceResourceTiming {initiatorType: "img", workerStart: 0, redirectStart: 0, ...}
  ▼ 6: PerformanceResourceTiming {initiatorType: "img", workerStart: 0, redirectStart: 0, ...}
```

Navigation Timing  
ResourceTiming

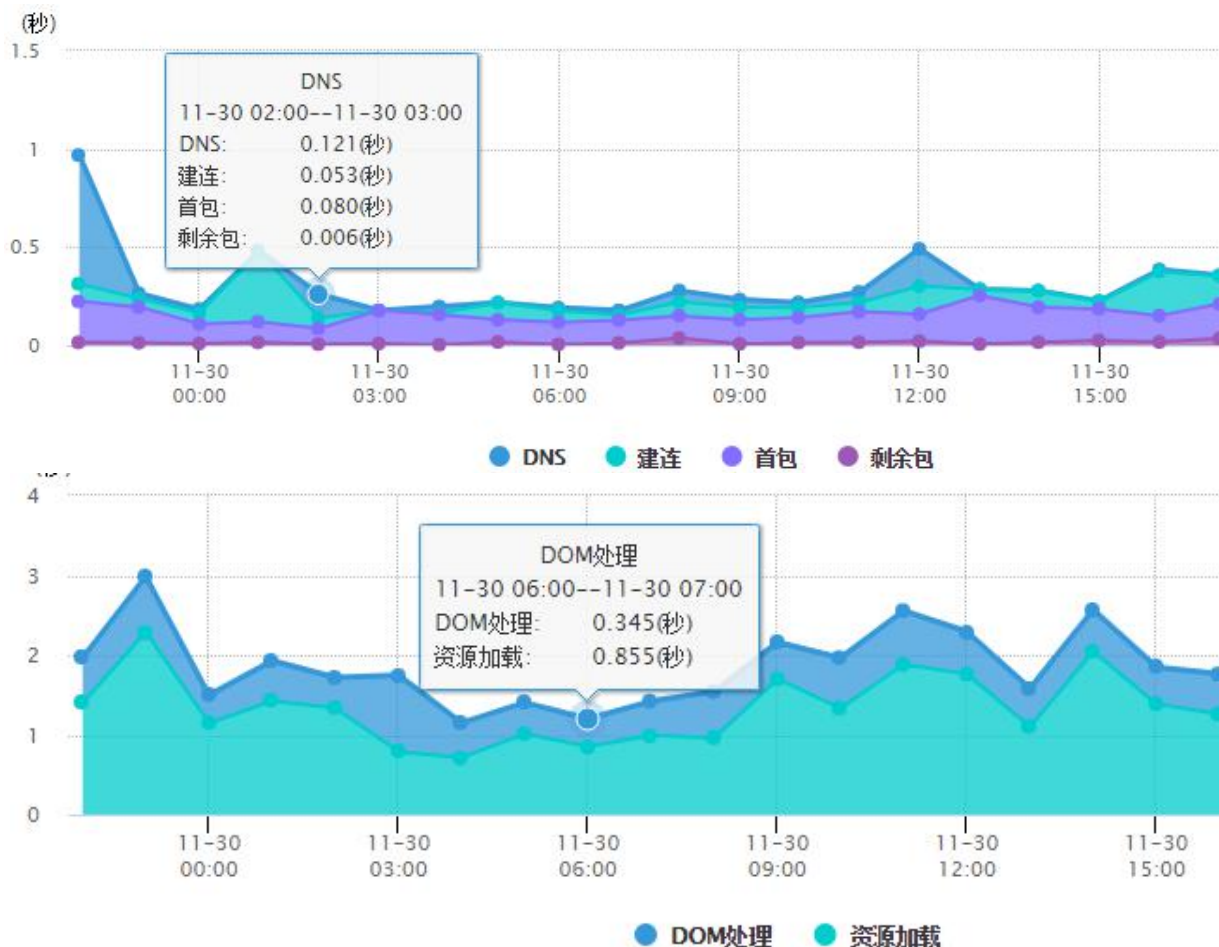
开始加载时间

Prom  
fo  
unlc

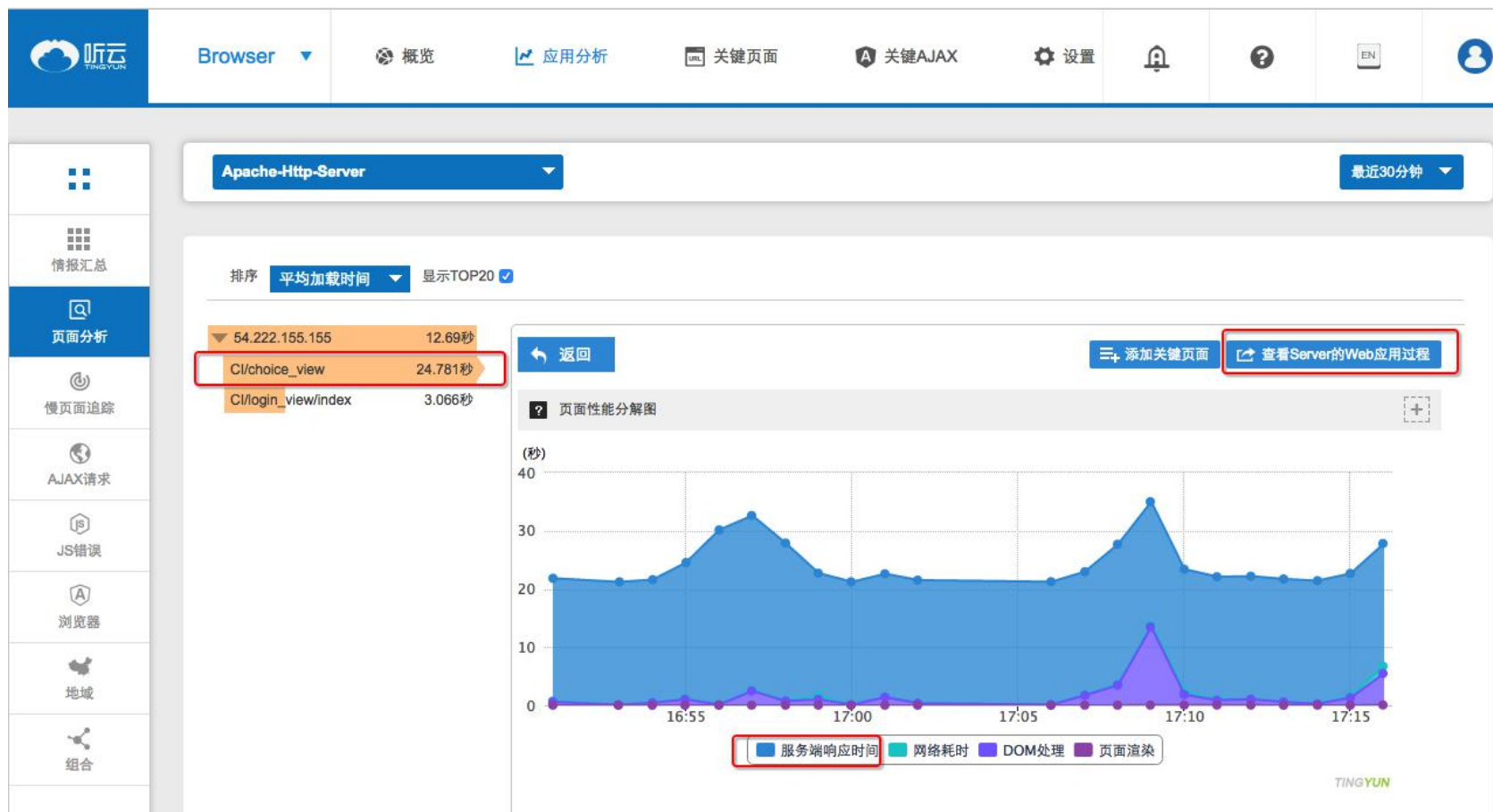
unloadEventStar  
unloadEventEn



## 浏览器->服务器溯源

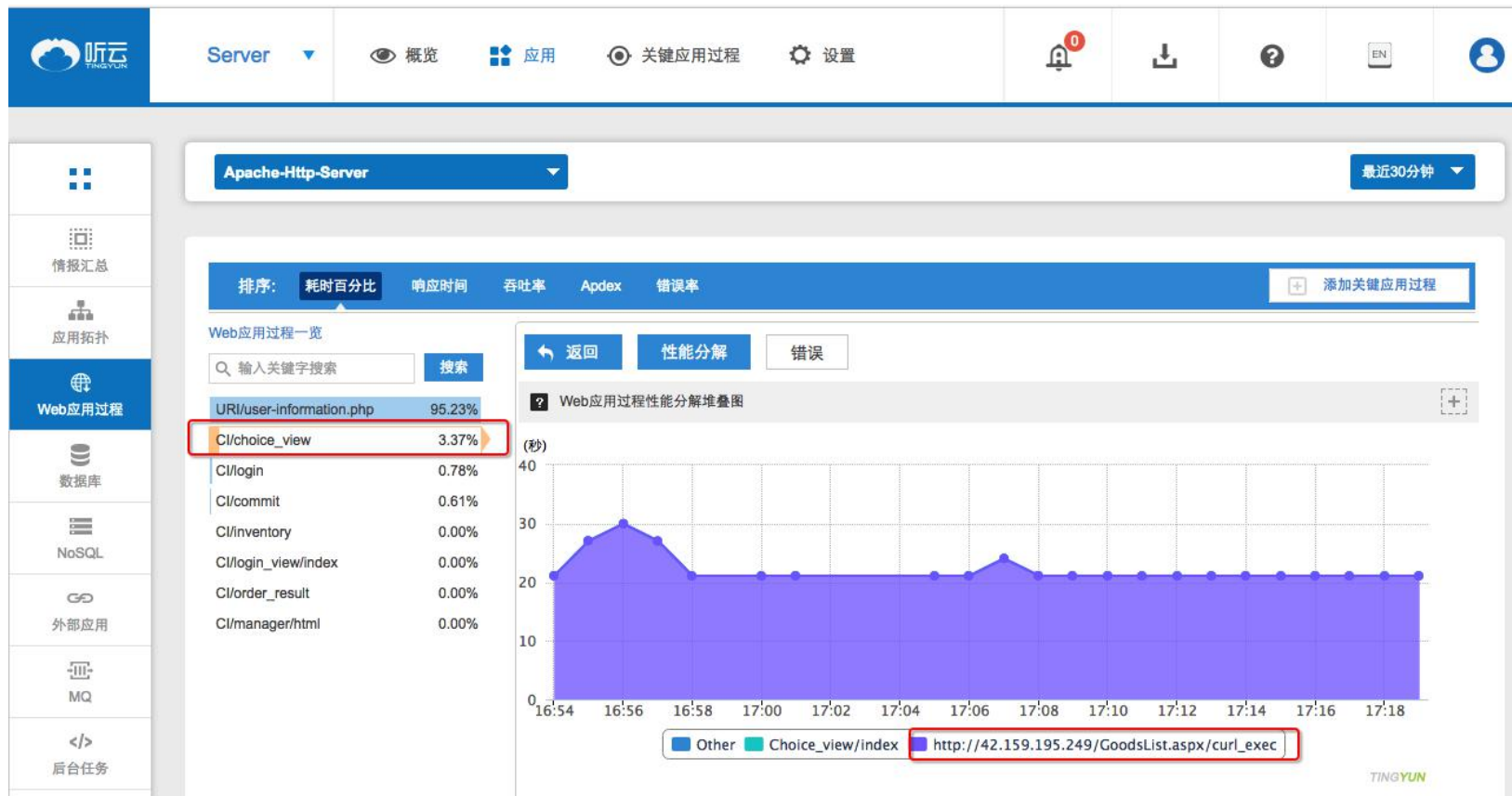


## 浏览器->服务器溯源

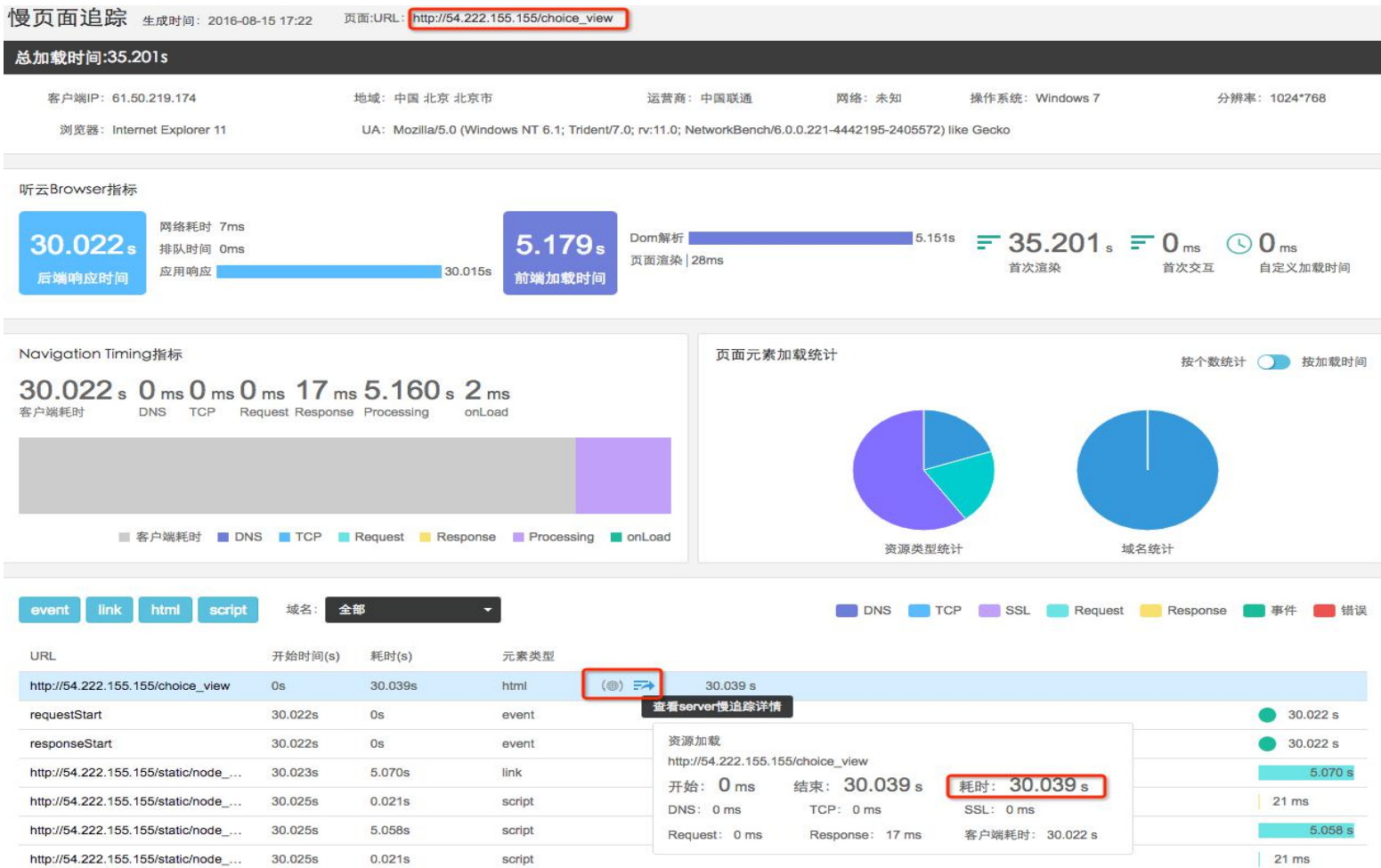




## 浏览器->服务器溯源



## 浏览器->服务器溯源



## 浏览器->服务器溯源

应用过程慢追踪

应用: [Apache-Http-Server](#)

应用过程: [Cl/choice\\_view](#)

追踪时间: 2016-08-15 17:22:04

服务器响应时间: 21.122 (s)

实例信息: [PHP:ip-10-0-1-54.cn-north-1.compute.internal](#)

共有 42 个应用追踪信息

摘要

追踪详情

相关SQL

展开所有

全部关闭

分类	持续时间(ms)	时间占比(%)	时间偏移量(ms)
▼ PHP.execute	21122	100.00	0
▼ require_once	≡ 21121	100.00	1
▼ call_user_func_array	≡ 21109	99.94	13
▼ Choice_view.index	≡ 21109	99.94	13
▼ Curl.simple_post	≡ 21108	99.93	13
▼ Curl.__call	≡ 21108	99.93	13
▼ call_user_func_array	≡ 21108	99.93	13
▼ Curl_simple_call	≡ 21108	99.93	13
▼ Curl.execute	≡ 21108	99.93	13
curl_exec	≡ 🌐 21108	99.93	13

### 请求信息

请求URL: /choice\_view

线程名称: pid-32066

HTTP响应: 200

referer: http://54.222.155.155/

user-agent: Mozilla/5.0 (Windows NT 6.1; Trident/7.0; rv:11.0; NetworkBench/6.0.0.22 1-3881978-2414601) like Gecko

### 请求参数

暂无请求参数

外部应用: [Inventory-Server](#) 实例信息: gartner-win

Web应用过程: [WebAction/ASP/GoodsList.aspx](#)

URL: [http://42.159.195.249/GoodsList.aspx](#)

耗时: 21037ms

耗时占比: 100%

应用层时间

© 2007-2016 北京基训网络股份有限公司

## 浏览器->服务器溯源

应用过程慢追踪

应用: [Inventory-Server](#)

应用过程: [ASP/GoodsList.aspx](#)

追踪时间: 2016-08-15 17:18:03

服务器响应时间: 21.017 (s)

实例信息: ASP.NET:gartner-win

共有 79 个应用追踪信息

摘要	追踪详情	相关SQL
展开所有	全部关闭	
分类		
		持续时间(ms) 时间占比(%) 时间偏移量(ms)
▼ DotNet.Execute		21017 100.00 0
HttpWebRequest.GetResponse		21017 100.00 0
StackTrace		
System.HttpWebRequest.GetResponse		

### 请求信息

请求URL: /GoodsList.aspx

线程名称:

HTTP响应: 200

referer:

Other

IP: 54.222.155.155

# App->服务器溯源

## App->服务器溯源

☒ 展示IP ☐

[+ 跳转到Web应用过程](#)

? 事务分解表格

代码段	性能分类	耗时百分比(%)	调用次数	平均响应时间(ms)
com.██████████uc.member.web.controller.UserGradeController/get...	Java	56.02	38448	16
HandlerInterceptor/preHandle	Spring	29.116	230772	1
com.██████████web.██████████patcherServlet/doDispatch	Java	4.714	33210	1
com.██████████uc.member.web.controller.UserGradeController/han...	SpringController	3.162	36602	1
Other	Java	3.096	692163	0
MongoDB-carrierswiflow-AGGREGATE	MongoDB	0.624	3465	2
HGETALL	Redis	0.469	192278	0
EXPIRE	Redis	0.4	209122	0

网络分解



# App->服务器溯源

追踪时间：2017-12-01 02:16:34

服务器响应时间：2.828 (s)

实例信息：JAVA:tm\_ ucDMZ001:20884

摘要

追踪详情

相关SQL

展开所有

全部关闭

分类		持续时间(ms)	时间占比(%)	时间偏移量(ms)
▼ ServletRequestListener.requestInitialized	≡	2828	100.00	0
▼ ServletRequestListener.requestInitialized	≡	2828	100.00	0

JsonpSupportInterceptor.preHandle

StackTrace		相关源文件：JsonpSupportInterceptor.java 行：40		
com. web.util.JsonpSupportInterceptor.preHandle		(JsonpSupportInterceptor.java:40)		
JsonpSupportInterceptor.preHandle		2815	99.54	1
JsessionInterceptor.preHandle		0	0.00	2816

# 监控实现原理

## 全栈溯源的核心：服务器端动态字节码增强

- Java

Instrumentation



bytecode

- PHP

Zend/Extensions



Opcache

```
public void xxoo() {  
    long startTime = System.currentTimeMillis();  
  
    try {  
        doXX();  
        doOO();  
  
        long endTime = System.currentTimeMillis();  
        long callTime = endTime - startTime;  
  
        APM.reportMetric("xxoo", callTime);  
    } catch (Exception ex) {  
        APM.reportError("xxoo",  
            ex.getMessage(),  
            ex.getStackTrace());  
  
        throw ex;  
    }  
}
```

1. 获取方法开始时间

2. 获取方法完成时间，并计算执行时间

3. 上报指标名及性能

4. 上报异常

## 全栈溯源的核心：服务器端动态字节码增强

```
public static void premain(String agentArgs, Instrumentation inst) {
    final ClassPool cp = ClassPool.getDefault();
    inst.addTransformer(new ClassFileTransformer() {
        @Override
        public byte[] transform(ClassLoader loader,
                                String className, Class<?> classBeingRedefined,
                                ProtectionDomain protectionDomain,
                                byte[] classfileBuffer) throws IllegalClassFormatException {
            if("xxoo/demo/Xxoo".equals(className)) {
                try {
                    String clazName = className.replace('/', '.');
                    cp.insertClassPath(new ByteArrayClassPath(clazName, classfileBuffer));
                    CtClass cc = cp.get(clazName);
                    CtMethod cm = cc.getDeclaredMethod("xxoo");
                    cm.insertBefore("APM.start();");
                    cm.insertAfter("APM.end();");
                    cm.insertAfter("System.out.println(\"xxoo invoked in: \"
                                + APM.elapsed() + \" ms.\");");
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
            return cc.toBytecode();
        }
    });
    return;
}
```

1. Classloading阶段自动注入监控代码

2. 启动命令行使用-javaagent参数上述代码实现自动嵌码

```
java -cp $CLASSPATH -javaagent:$APM_AGENT_PATH/apm-agent.jar
xxoo.demo.XxooMain
```

## 全栈溯源的核心：服务器端动态字节码增强

- 可动态增强字节码的开源框架：asm, javassist
- 指标的采集均可通过函数/方法的拦截来实现

eg: javax.servlet.http.HttpServlet.service(req, resp)

✓服务响应时间 ✓执行异常 ✓修改HTTP头

java.sql.Statement.executeQuery(sql)

✓SQL执行时间 ✓执行异常 ✓上下文SQL

org.apache.http.client.HttpClient.execute(req)

✓HTTP响应时间 ✓执行异常 ✓修改HTTP头

## 浏览器端Js埋码

- Navigation-timing/Resource-timing接口采集主要监控数据  
页面及元素的DNS、TCP、SSL、DOM渲染等时序数据
- AJAX单独埋点采集  
hook XMLHttpRequest.open/send函数，采集AJAX请求的  
响应时间、回调时间等
- 监听特定事件  
onerror，采集js错误数据



## 全栈溯源（不同端）如何关联？

- 浏览器端/App/服务器端自动嵌码
- 服务器端，ThreadLocal或异步Context关联前后端请求及调用
- 服务器端拦截HttpServlet或HttpClient，修改HTTP头
- 拦截JSP/PHP编译过程，修改Response输出内容（<head>/<body>）
- (Ajax)拦截XmlHttpRequest，修改HTTP头
- App/浏览器→Trace ID/ReqId→服务器

## 小结及其它

- 服务器端/浏览器/App端自动嵌码采集监控指标
- 打通不同端的监控是重点
- Server→Server(DB/redis/MQ/API gateway/微服务)
- 调用链、根因分析

GIAC | 全球互联网架构大会  
GLOBAL INTERNET ARCHITECTURE CONFERENCE

GIAC

全球互联网架构大会

GLOBAL INTERNET ARCHITECTURE CONFERENCE



扫码关注GIAC公众号

2017.thegiac.com