

ArchData

技术峰会成都站

主办方：



2018年1月27日成都菁蓉国际广场3W COFFICE 蓉漂茶馆



Skywalking

彭勇升

Skywalking PPMC

天源迪科中台研发总监

发展史

2015

- 创建项目
- 提供SDK通过手工埋点的方式抓取链路信息，链路的spanId的生成规则参考阿里的鹰眼

2016

- 使用byte-buddy实现自动埋点
- 自动埋点的实现采用插件的机制区分不同的框架和不同的版本
- 探针重构：优化性能，移除鹰眼的spanId的生成规则

2017

- 探针与后端的协议重构：为后端分析而生
- 后端重构：提供链路数据的分析和分类聚合
- 界面重构：提供一个APM系统最基本的能力（应用拓扑图、链路展示、服务关系依赖、JVM统计指标）

2018

- 界面重构：比肩甚至超越商业APM系统的界面
- 探针：其他语言的支持（Php, Pthon, Go...）
- 后端：通过GraphQL将前后端接口协议化，更丰富的统计指标、预警、历史数据自动清理、路由节点推测，提升性能和稳定性

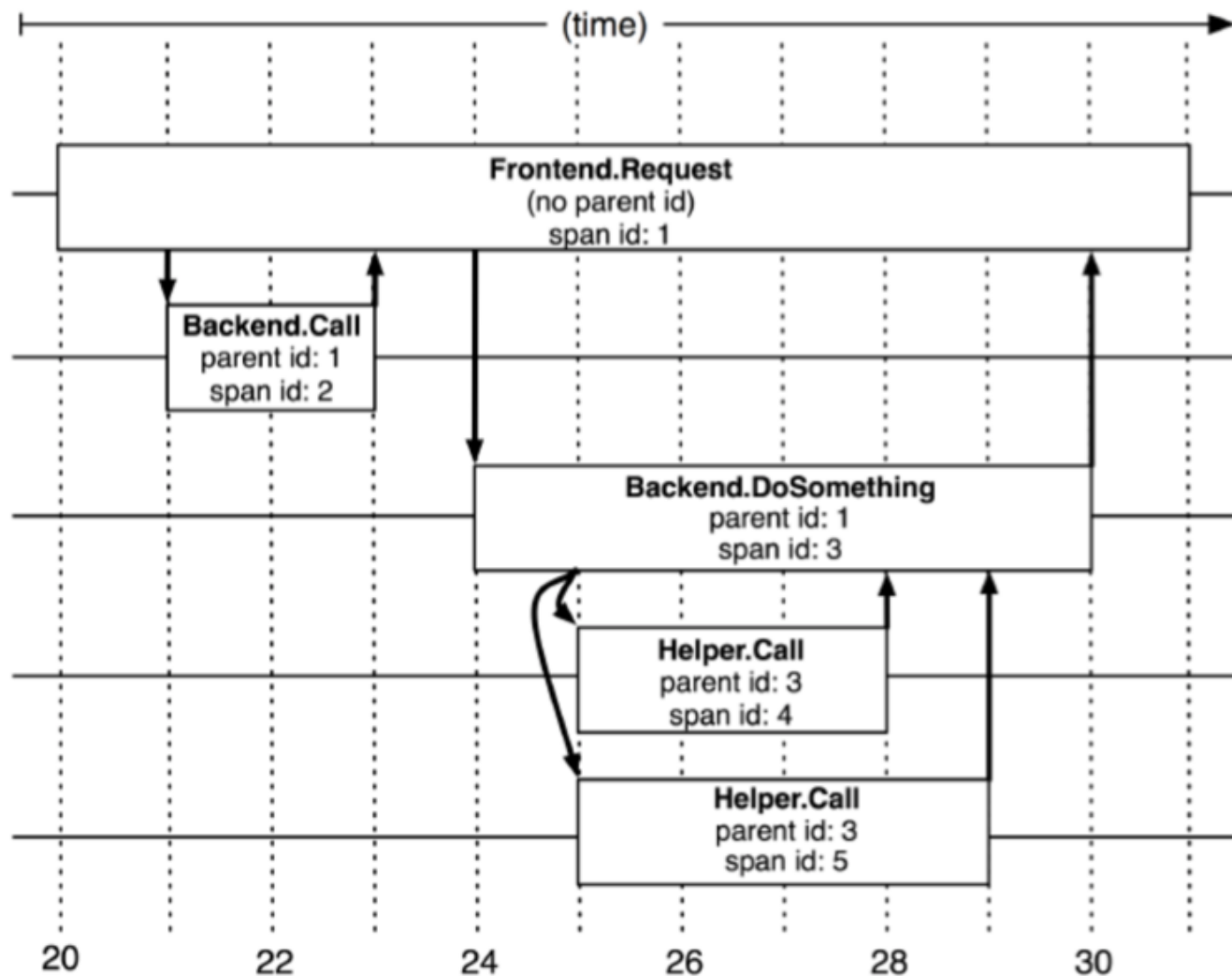


构建自己的分布式追踪程序



理论和原理

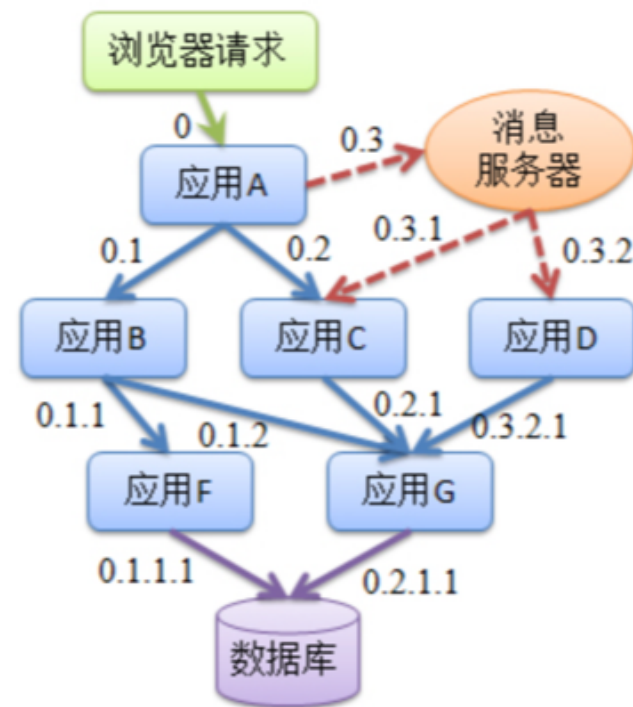
理论基础：Google Dapper Paper



鹰眼 != Dapper

It is just an implementation

绝大多数的追踪系统，并不使用这种编号方式



探针工作原理 (1)

- Application codes

```
PreparedStatement updateSales = con.prepareStatement(  
    "UPDATE COFFEES SETSALES = ? WHERE COF_NAME LIKE ? ");  
updateSales.setInt(1, 75);  
updateSales.setString(2, "Colombian");  
updateSales.executeUpdate();
```

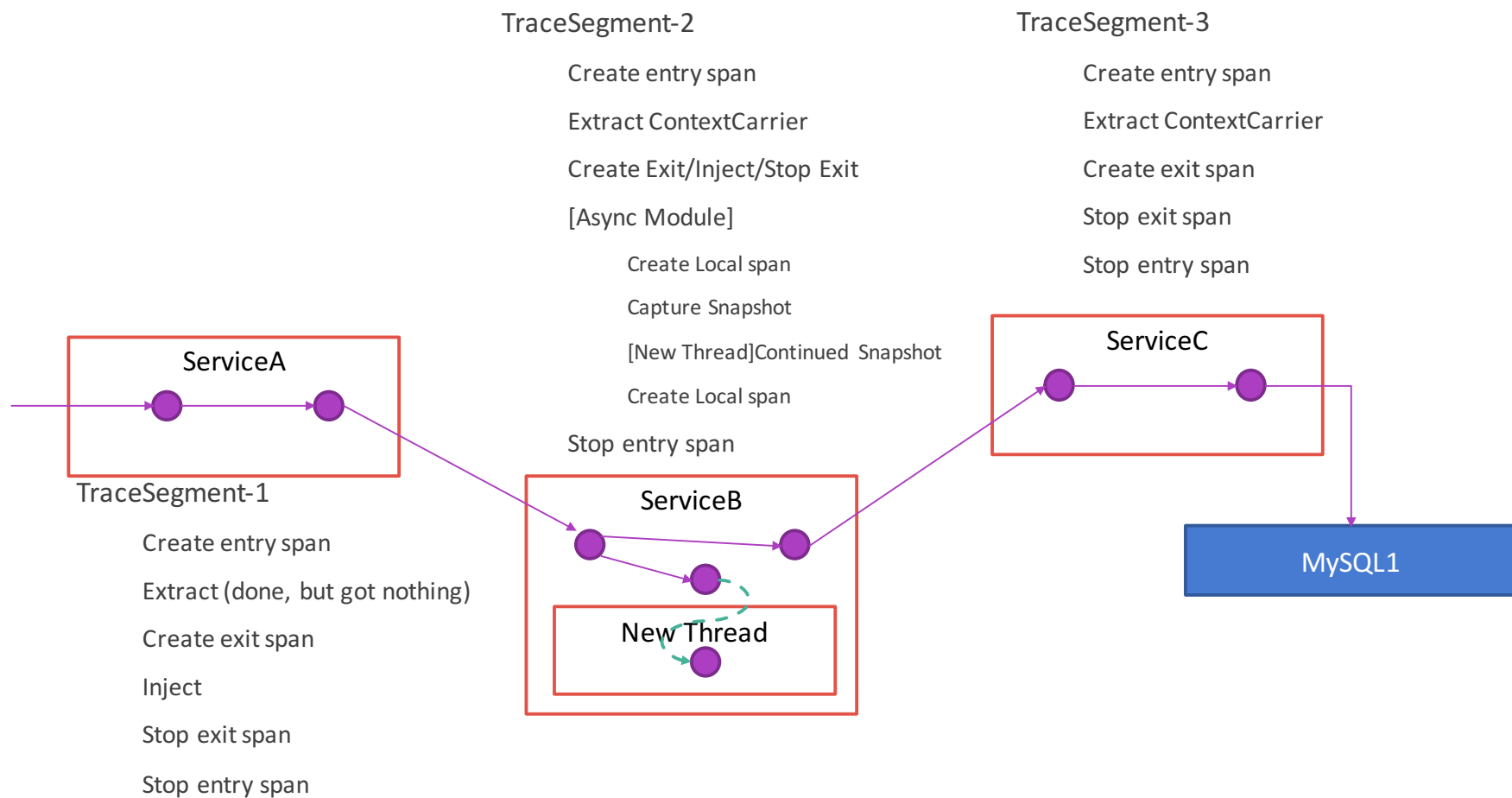
- **Running with Instrumentation, Pseudocode Only**

- Auto-instrumentation mechanism

- AOP(Aspect Oriented Programming) in byte-code level.
- Manipulate codes at runtime.

```
PreparedStatement updateSales = con.prepareStatement(  
    "UPDATE COFFEES SETSALES = ? WHERE COF_NAME LIKE ? ");  
● tracer.cacheSQL("UPDATE COFFEES SETSALES = ? WHERE COF_NAME LIKE ? ");  
    updateSales.setInt(1, 75);  
● tracer.cacheDbParam(1, 75);  
    updateSales.setString(2, "Colombian");  
● tracer.cacheDbParam(2, "Colombian");  
    Span span = tracer.createSpan().start();  
    updateSales.executeUpdate();  
● span.tag(Tags.Basic.TYPE, "Database");  
    span.tag(Tags.SQL, tracer.getCachedSQL());  
    span.tag(Tags.SQL_PARAMETERS, tracer.getCachedDbParams());  
    span.stop();
```

探针工作原理 (2)





实践出真知

插件开发-切面

```
public class RealCallInstrumentation extends ClassInstanceMethodsEnhancePluginDefine {
```

```
    private static final String ENHANCE_CLASS = "okhttp3.RealCall";
```

需要增强的类

```
    private static final String INTERCEPT_CLASS = "org.apache.skywalking.apm.plugin.okhttp.v3.RealCallInterceptor";
```

插码的实现类

```
    @Override protected ClassMatch enhanceClass() { return NameMatch.byName(ENHANCE_CLASS); }
```

```
    @Override protected ConstructorInterceptPoint[] getConstructorsInterceptPoints() {
```

定义拦截点：构造函数

```
        return new ConstructorInterceptPoint[] {
```

```
            new ConstructorInterceptPoint() {
```

```
                @Override public ElementMatcher<MethodDescription> getConstructorMatcher() { return any(); }
```

匹配所有构造

```
                @Override public String getConstructorInterceptor() { return INTERCEPT_CLASS; }
```

```
            }
```

```
        };
```

```
    }
```

```
    @Override protected InstanceMethodsInterceptPoint[] getInstanceMethodsInterceptPoints() {
```

定义拦截点：方法

```
        return new InstanceMethodsInterceptPoint[] {
```

```
            new InstanceMethodsInterceptPoint() {
```

```
                @Override public ElementMatcher<MethodDescription> getMethodsMatcher() { return named("execute"); }
```

匹配名称为execute的方法

```
                @Override public String getMethodsInterceptor() { return INTERCEPT_CLASS; }
```

匹配类名

```
                @Override public boolean isOverrideArgs() { return false; }
```

```
            },
```

```
            new InstanceMethodsInterceptPoint() {
```

```
                @Override public ElementMatcher<MethodDescription> getMethodsMatcher() {
```

```
                    return named("enqueue").and(takesArguments( length: 1));
```

```
                }
```

```
                @Override public String getMethodsInterceptor() {
```

```
                    return "org.apache.skywalking.apm.plugin.okhttp.v3.EnqueueInterceptor";
```

```
                }
```

```
            @Override public boolean isOverrideArgs() { return false; }
```

```
        }
```

```
    };
```

```
}
```

插件开发-插码（前置）

```
@Override public void beforeMethod(EnhancedInstance objInst, Method method, Object[] allArguments,
    Class<?>[] argumentsTypes, MethodInterceptorResult result) throws Throwable {
    Request request = (Request)objInst.getSkyWalkingDynamicField();

    ContextCarrier contextCarrier = new ContextCarrier();
    HttpUrl requestUrl = request.url();
    AbstractSpan span = ContextManager.createExitSpan(requestUrl.uri().getPath(), contextCarrier, remotePeer: requestUrl.host() + ":" + requestUrl.port());
    span.setComponent(ComponentsDefine.OKHTTP);
    Tags.HTTP.METHOD.set(span, request.method());
    Tags.URL.set(span, requestUrl.uri().toString());
    SpanLayer.asHttp(span);

    Field headersField = Request.class.getDeclaredField( name: "headers");
    Field modifiersField = Field.class.getDeclaredField( name: "modifiers");
    modifiersField.setAccessible(true);
    modifiersField.setInt(headersField, headersField.getModifiers() & ~Modifier.FINAL);

    headersField.setAccessible(true);
    Headers.Builder headerBuilder = request.headers().newBuilder();
    CarrierItem next = contextCarrier.items();
    while (next.hasNext()) {
        next = next.next();
        headerBuilder.add(next.getHeadKey(), next.getHeadValue());
    }
    headersField.set(request, headerBuilder.build());
}
```

创建Exit Span，设置关键属性

设置其他属性：

1. 组件类型
2. 所属层
3. 非重要Tag和Log

传递上下文

插件开发-插码（后置和异常）

```
@Override
public Object afterMethod(EnhancedInstance objInst, Method method, Object[] allArguments,
    Class<?>[] argumentsTypes, Object ret) throws Throwable {
    Response response = (Response)ret;
    int statusCode = response.code();

    AbstractSpan span = ContextManager.activeSpan();
    if (statusCode >= 400) {
        span.errorOccurred();
        Tags.STATUS_CODE.set(span, Integer.toString(statusCode));
    }

    ContextManager.stopSpan();

    return ret;
}

@Override public void handleMethodException(EnhancedInstance objInst, Method method, Object[] allArguments,
    Class<?>[] argumentsTypes, Throwable t) {
    AbstractSpan abstractSpan = ContextManager.activeSpan();
    abstractSpan.errorOccurred();
    abstractSpan.log(t);
}
```

获取当前Span

根据状态标记是否调用错误

结束当前Span

异常捕获标记是否调用错误



探针和后端的协议详解

探针与后端的协议详解 (1)

```
[
{
  "gt": [[230150, 185809, 24040000]], //trace_id 链路编码
  "sg": {
    "ts": [230150, 185809, 24040000], //segment_id 一个应用中的一个实例在链路中产生的编号
    "ai": -1, //应用ID, 应用启动注册时产生
    "ii": 2, //实例ID, 应用启动注册时产生
    "rs": [], //链路的上下文信息, 头节点, 所以为空
    "ss": [ //一个埋点 (手动埋点或者自动埋点)
      {
        "si": 1, //埋点编号
        "tv": 1, //埋点类型 (手动、入口、出口)
        "lv": 1, //层 (RPCFramework, Database, Http)
        "ps": 0, //上级埋点编号
        "st": 1501858094526, //开始时间
        "et": 1501858097004, //结束时间
        "ci": 3, //组件编号 (Tomcat, Mysql, Dubbo...)
        "cn": "", //组件名称
        "oi": 0, //埋点的服务ID
        "on": "org.skywaking.apm.testcase.dubbo.services.GreetService.doBusiness()", //埋点的服务名
        "pi": 0, //地址注册的ID, 调用抓取后注册
        "pn": "172.25.0.4:20880", //地址
        "ie": false, //是否发生错误, 是否有Exception
        "to": [ //与链路分析关系较小的TAG, SQL也在其中
          {
            "k": "url",
            "v": "rest://172.25.0.4:20880/org.skywaking.apm.testcase.dubbo.services.GreetService.doBusiness()"
          }
        ],
        "lo": [] //日志, 与链路分析完全无关, 只作为展示使用 | 手动埋点时可以任意追加, kv的形式, 例如收集一些业务相关的信息
      }
    ],
  },
}
```

探针与后端的协议详解 (2)

```
[
{
  "gt": [[230150, 185809, 24040000]], //trace_id 链路编码, 与调用方相同
  "sg": {
    "ts": [137150, 185809, 48780000], //segment_id, 新产生
    "ai": 2,
    "ii": 3,
    "ss": [
      {
        "si": 0,
        "tv": 0,
        "lv": 2,
        "ps": -1,
        "st": 1501858094726,
        "et": 1501858096804,
        "ci": 3,
        "cn": "",
        "oi": 0,
        "on": "org.skywaking.apm.testcase.dubbo.services.GreetService.doBusiness()",
        "pi": 0,
        "pn": "",
        "ie": false,
        "rs": [
          {
            "pts": [230150, 185809, 24040000], //上级的segment_id 一个应用中的一个实例在链路中产生的编号
            "pii": 2, //上级的实例编号
            "psp": 0, //上级的埋点编号span_id
            "psi": 0, //上级的服务编号(org.skywaking.apm.testcase.dubbo.services.GreetService.doBusiness()注册后的ID)
            "psn": "org.skywaking.apm.testcase.dubbo.services.GreetService.doBusiness()", //上级的服务名
            "ni": 0, //上级调用时使用的地址注册后的ID
            "nn": "172.25.0.4:20880", //上级的地址
            "eii": 2, //入口的实例编号
            "esi": 0, //入口的服务编号
            "esn": "/dubbox-case/case/dubbox-rest", //入口的服务名词
            "rn": 0 //调用方式 (CrossProcess, CrossThread)
```



5.0功能预览

5.0功能预览 (看板)

The screenshot displays the SkyWalking 5.0 dashboard interface. The top navigation bar includes the SkyWalking logo and a 'Dashboard' link. The main content area is divided into several sections. On the left, there's a sidebar with a menu. The central part of the dashboard features a 'Time Range' selector and a 'Reloading every' dropdown menu, with an 'Apply' button below them. Below this, there's a 'Slow Service' section with a list of names: Eric Hernandez, Scott Davis, Carol Hall, Gary Young, Ruth Anderson, Timothy Walker, and Sarah Jones. To the right, there's a 'Last 5 minutes' section with a refresh icon and a notification bell icon showing 3 alerts. A modal window titled 'Application Alert (3)' is open, displaying a list of alerts: 'Appliction A error' on 2017-08-09, 'Appliction A error' on 2017-08-08, and 'Appliction A error' on 2017-08-07. Below the list, there's a 'More Application Alert' link. The bottom of the dashboard shows a table with columns for 'Name', 'Value', and 'Unit'. The first row shows 'Sarah Jones' with a value of '853'. The second row shows 'Brian Johnson' with a value of '2246'.

Sw SkyWalking

Dashboard

Last 5 minutes

Application Alert (3)

Server Alert (3)

Time Range: Start date ~ End date

Reloading every : [dropdown]

Apply

Min 11%

Slow Service

Name

Eric Hernandez

Scott Davis

Carol Hall

Gary Young

Ruth Anderson

Timothy Walker

Sarah Jones

853

Brian Johnson

2246

Appliction A error

2017-08-09

Appliction A error

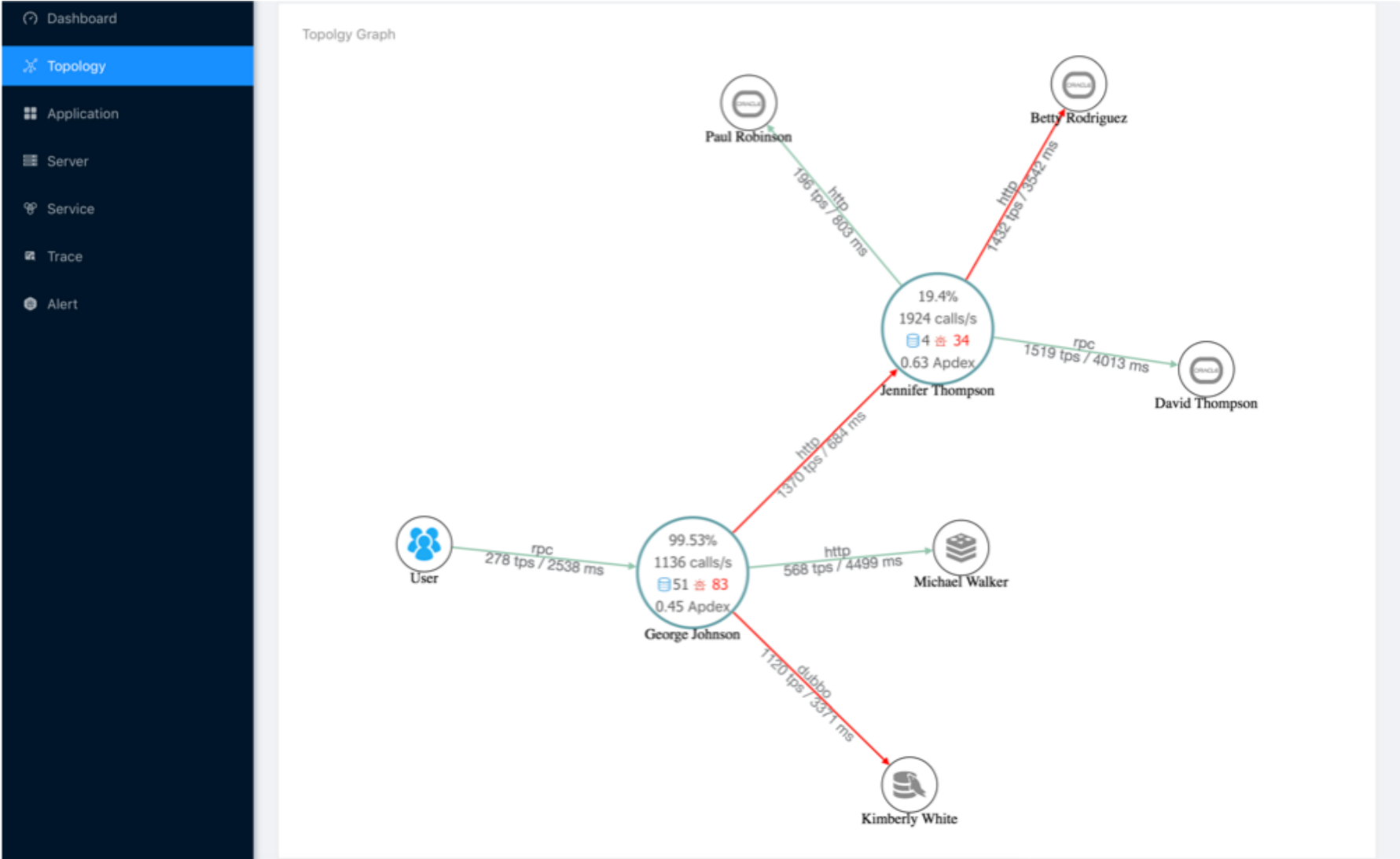
2017-08-08

Appliction A error

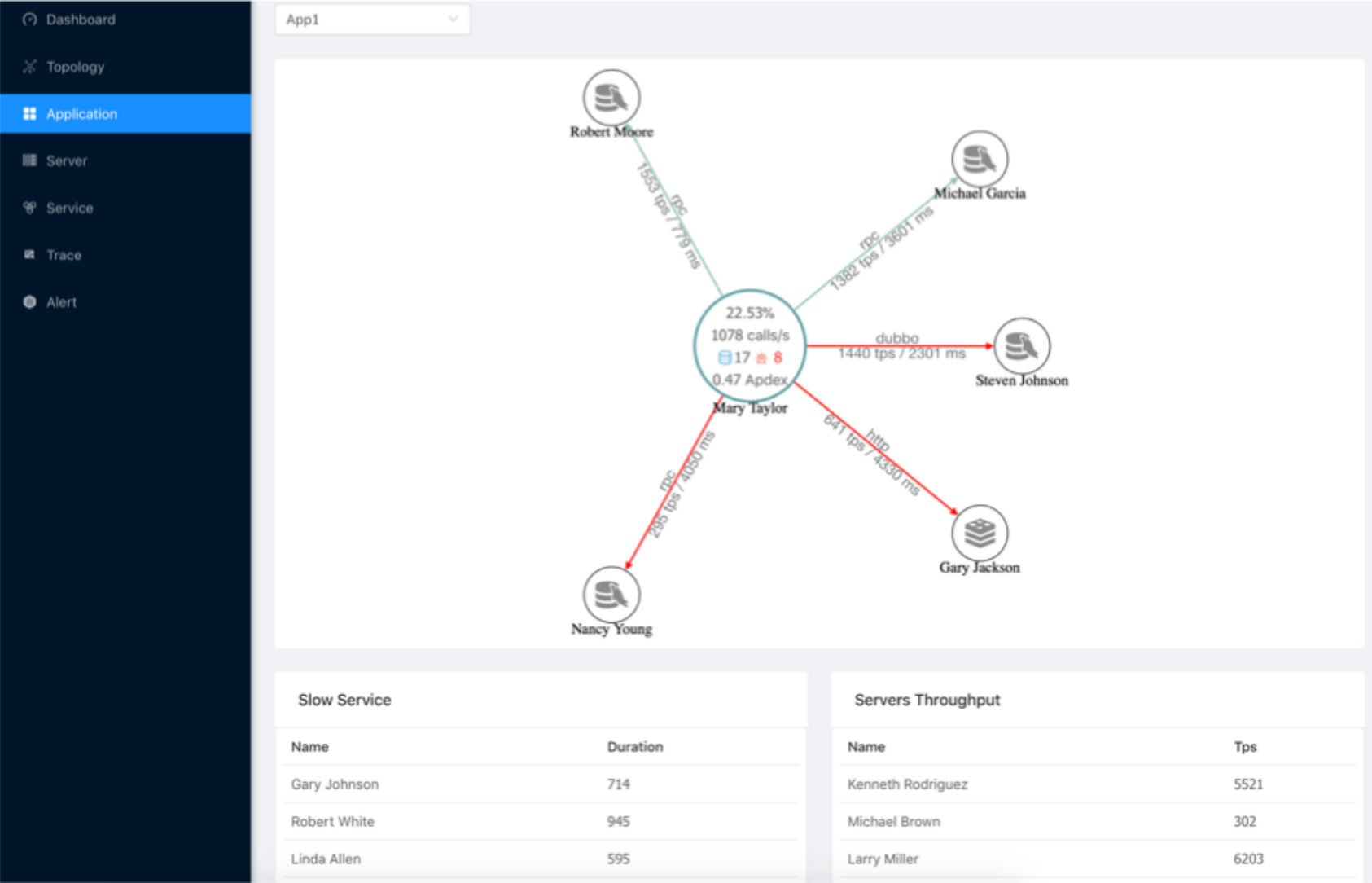
2017-08-07

More Application Alert

5.0功能预览（全局拓扑图）



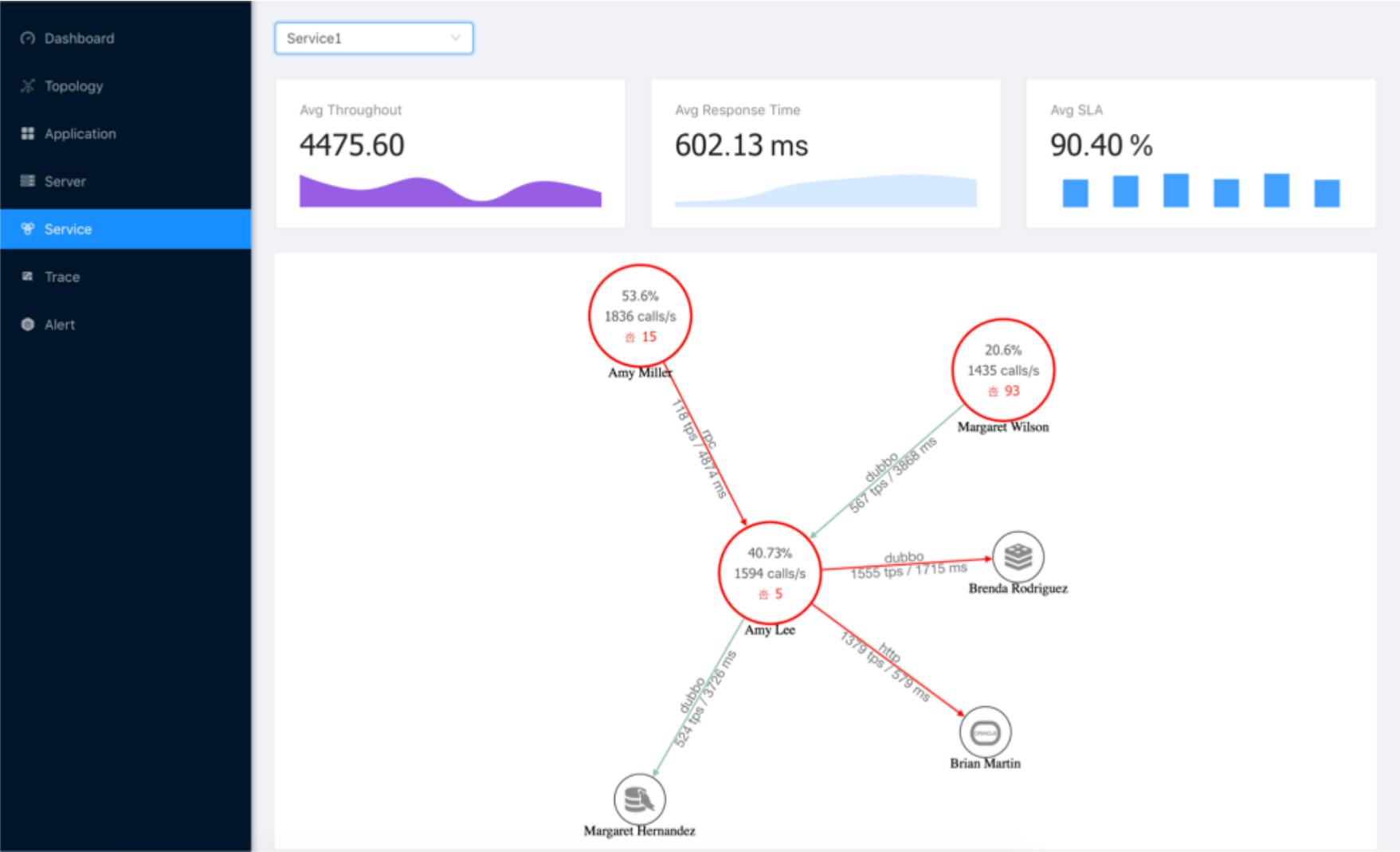
5.0功能预览（应用）



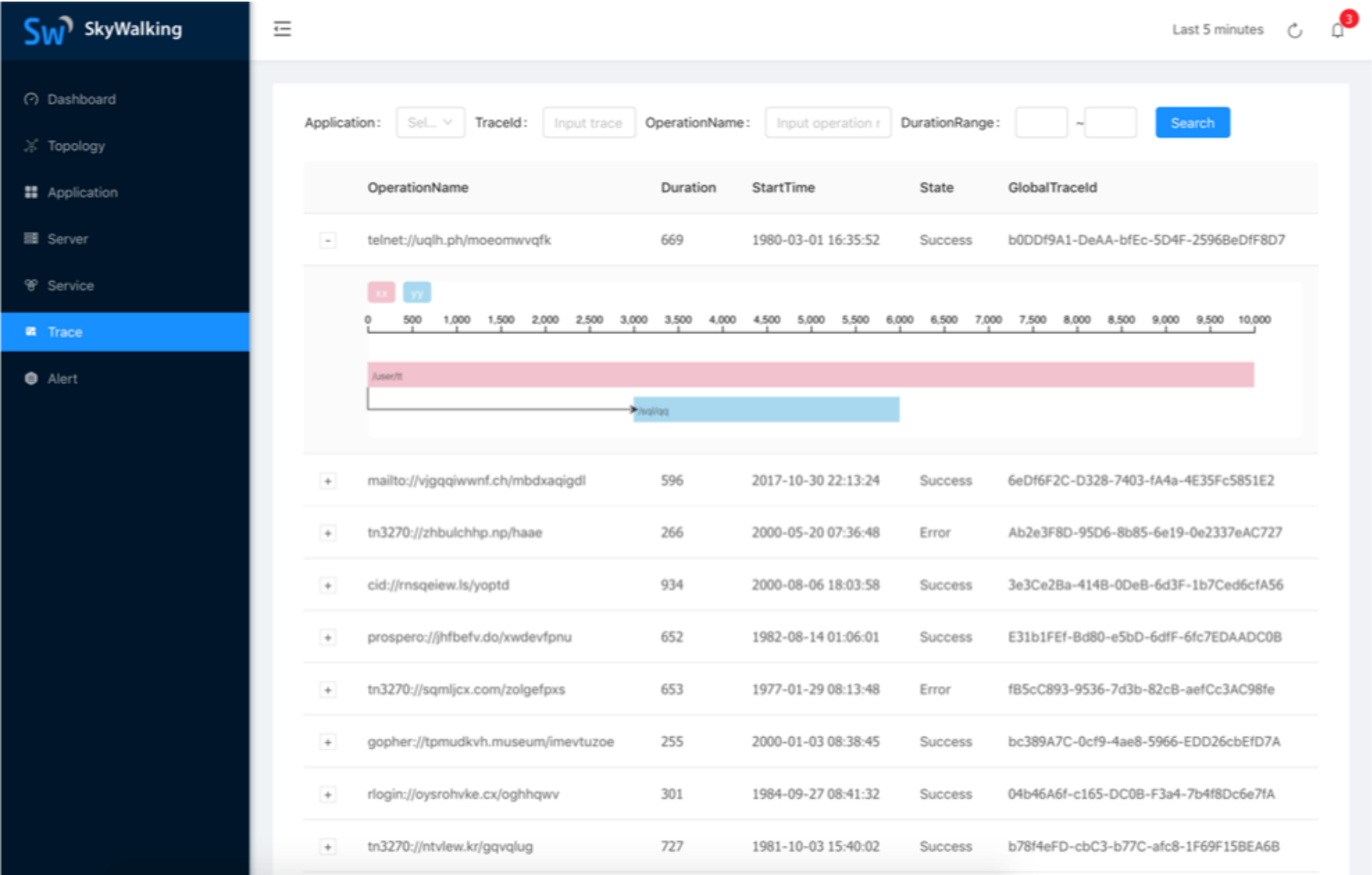
5.0功能预览（实例）



5.0功能预览（服务）



5.0功能预览（链路查询）



5.0功能预览（预警）

Sw SkyWalking

Dashboard

Topology

Application

Server

Service

Trace

Alert

Alert List

Search...

Content	Start Time	Alert Type
&Dz@iCeeZbEC6[@]JH\$Y	2007-12-17 08:49:51	SERVICE
wnFr^FZ%tTYX!nmkVYLV	2008-03-05 07:52:38	APPLICATION
twBPgQ!27L\$72Y#ZPZ1Z	1974-12-25 06:45:25	APPLICATION
t)#@@3yMK*nRW&reDYyZ	1996-09-18 19:26:11	APPLICATION
9cqk#suWGRMg%cVjT#W3	1985-03-13 15:12:28	APPLICATION
eF*mAprXJkb@KV9e%T4J	1977-07-26 05:12:39	SERVICE
z2!7^mGoF@0D8LTExfji	2002-02-15 05:55:37	SERVER
4Ddg@2O5vfulB#z7[A(\$	2012-01-16 12:44:48	SERVICE
WYiqVeV7WRu2JvMOyZC9	2013-03-20 22:53:01	SERVER
M@O@&QQ#awbWy#JRH0(a	1994-11-22 14:30:38	SERVER

<

1

2

3

4

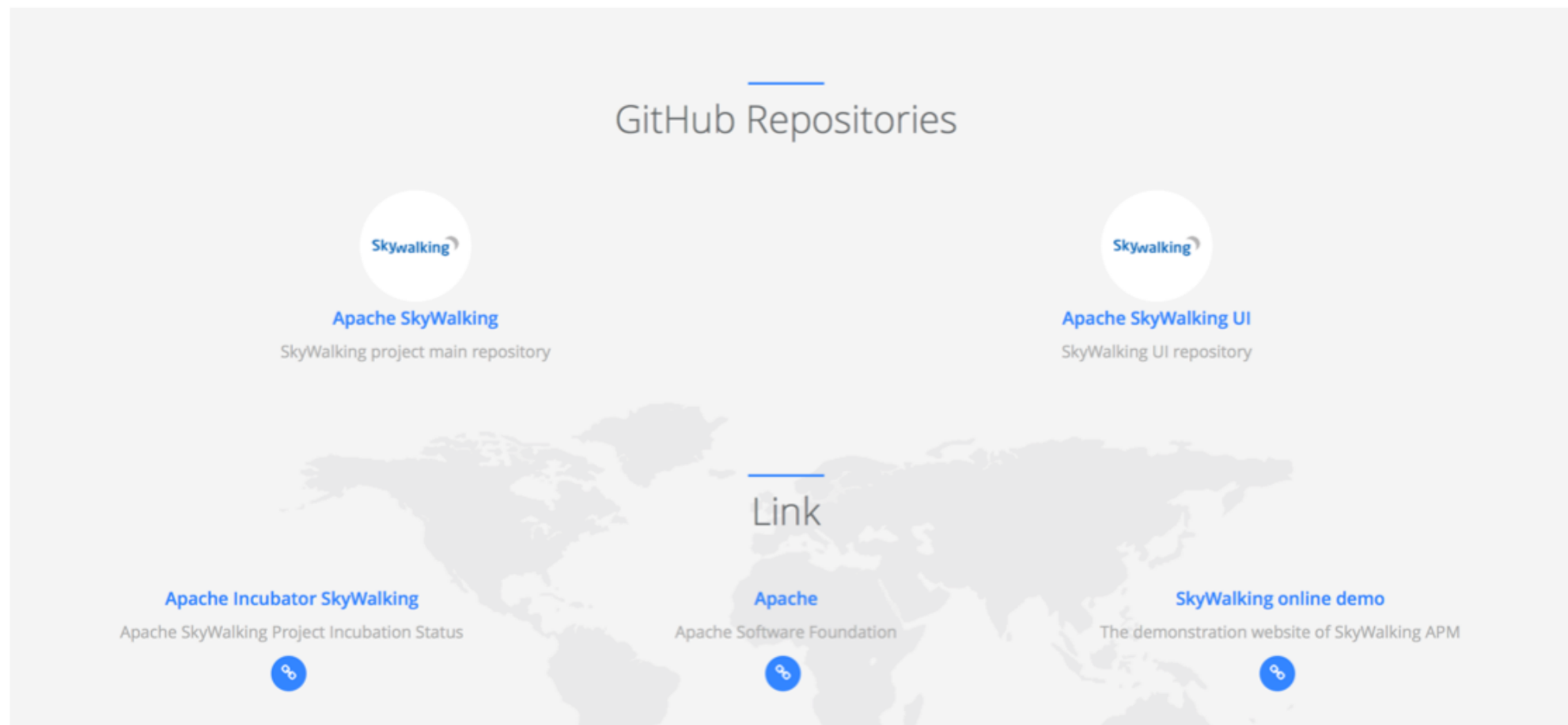
5

...

10

>

下载



横跨全年的品牌传播与实效营销盛宴

同城合纵

年度技术峰会+ArchData巡回
覆盖全国华北、华东、华南、西南、西北地理区域中心城市
200+媒体传播，受众超千万

年度技术峰会
北京、上海、成都

杭州技术沙龙

广州技术沙龙

深圳技术沙龙

西安技术沙龙

南京技术沙龙

厦门技术沙龙

北京技术沙龙

成都技术沙龙
上海技术沙龙

年终技术峰会

全域连横

年度技术峰会

成都技术沙龙

走进大数据案例企业

走进电商案例企业

走进金融案例企业

科技厂商探访

企业痛点诊断

旅游交流活动

成都技术沙龙

年终技术峰会

以上海为例

年度技术峰会+一杆到底，
围绕北京/上海/成都，经济标杆城市，步步为营，深耕用户
多重形式，复数传播，挖掘地域商务价值

合纵连横，在中国开发者群体中缔造品牌营销奇迹



中生代技术
FRESHMAN TECHNOLOGY

ArchData技术峰会
聚焦人工智能、大数据、基础架构、区块链等
前沿课题

中生代技术提供咨询内训服务
技术架构，研发管理，敏捷开发，大数据
微服务，AI，机器学习等

中生代技术提供人才服务
对接研发主管，内推精准人才