

掌握字节码插桩技术的核心功能

概要：

- 1、Javaagent基本使用
- 2、javassist 核心API
- 3、插桩技术应用实践

讲师介绍：

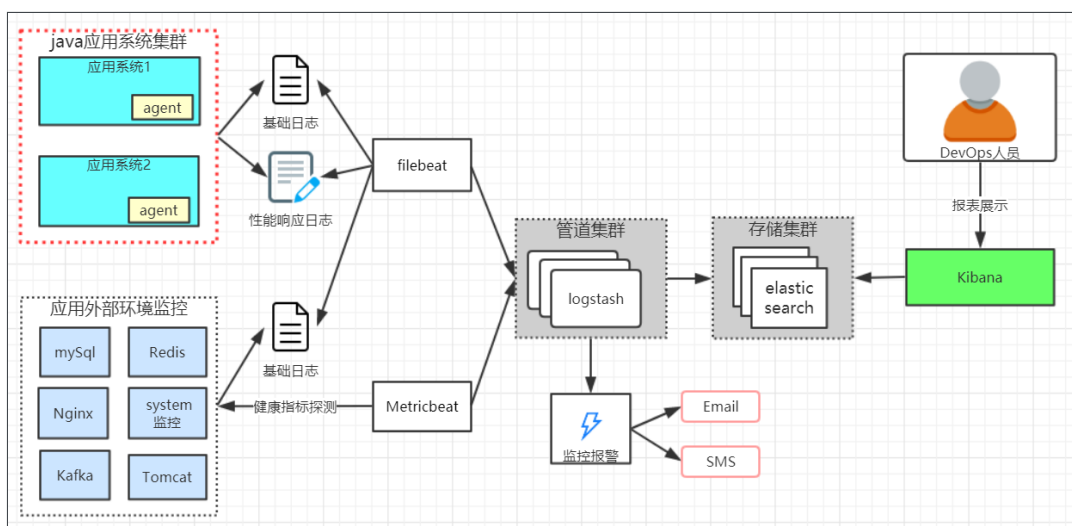


主讲老师

代号：鲁班 曾广炜

多年的互联网技术开发和管理经验，曾任云猴网架构师，参与多个大型互联网平台的搭建，擅长API接口设计。目前正在研究通过工具解决团队编码效率的问题。
QQ:2877438881

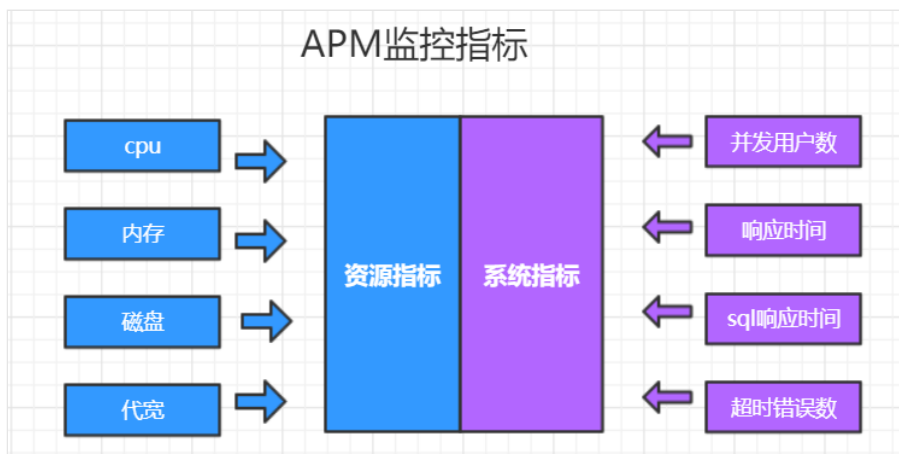
项目架构回顾：



插桩技术掌握对图灵APM项目的意义：

APM 整体实现 分为 数据采集、传输、存储、图表展示与报警。当中采集可以分为资源指标和系统指标两类。一般监控系统调用操作系统命令可以轻松获得资源指标。而内部系统的指标要么就是应用自己采集数据发送数据监控系统，要么引入监控系统提供的JAR

来修改，无论哪种都对应用造成了侵入，不利于实施。字节码插桩技术则可以解决监控系统侵入这一问题。



字节码插桩技术怎么实现呢?我们只需掌握两个技术即可：

1. 插桩的入口javaagent。
2. 字节码修改工具javassist。

一、Javaagent基本使用

1. javaagent简介与应用场景
2. javaagent 如何使用
3. javaagent 执行机制

javaagent简介

javaagent 是java1.5之后引入的特性，其主要作用是在class 被加载之前对其拦截，已插入我们的监听字节码

javaagent 应用场景

监控、代码覆盖率分析
JProfiler、应用破解

javaagent 使用

javaagent 其实就是一个jar 包，通过-javaagent:xxx.jar 引入监控目标应用。那这个jar 和普通的jar 有什么区别？

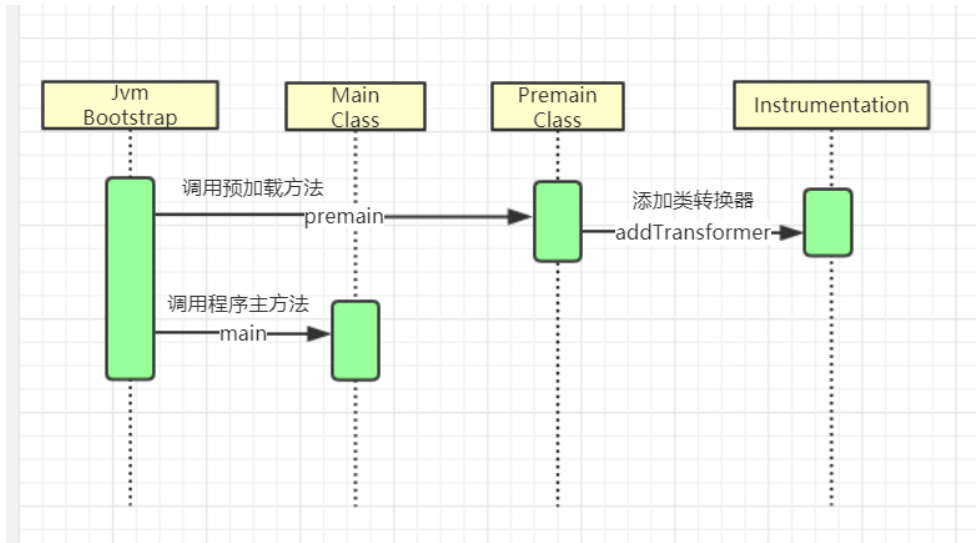
	运行时JAR包	Javaagent JAR包
入口方法名称	Main	premain
Maninfe.MF 主要参数	Main-class:	Premain-class:
启动参数	java -jar xxx.jar	-javaagent:xxx.jar
执行顺序		先
是否独立启动	是	否

javaagent 使用演示

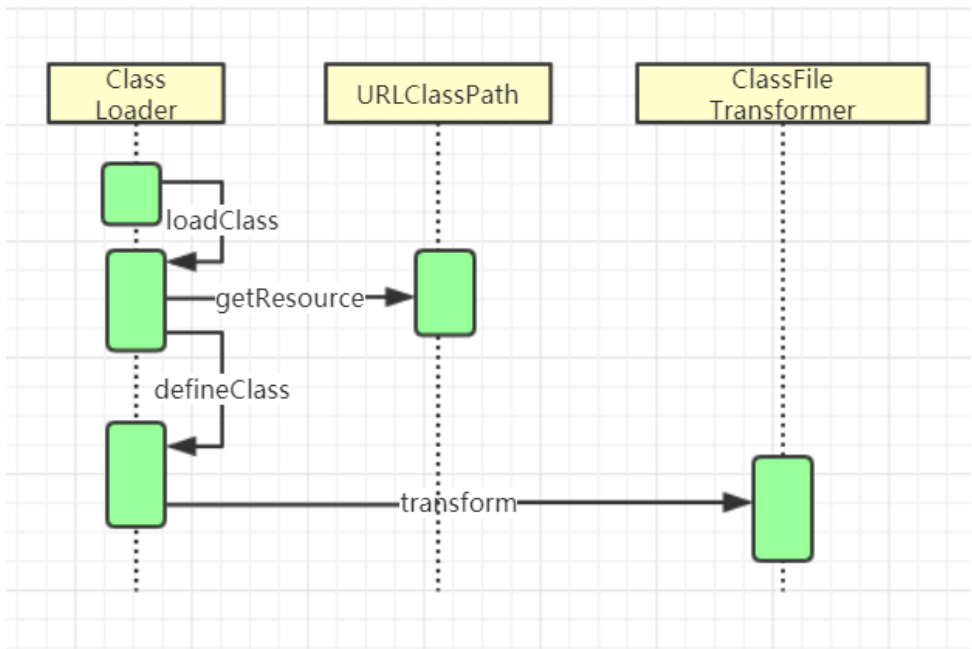
☐ 演示静态装载agent

javaagent 底层流程：

javaagent 装载时序图（premain）：



Class 装载时序图：



javaagent jar 包MANIFEST.MF 配置参数

```
#动态agent 类
Agent-Class: com.tuling.javaagent.MyAgent
#agent 依赖包逗号分割
Boot-Class-Path: javassist-3.18.1-GA.jar
#是否允许重复装载
Can-Redefine-Classes: true
#静agent 类
Premain-Class: com.tuling.javaagent.MyAgent
```

二、javassist 核心API

1. javassist 简介与应用场景
2. javassist 如何使用
3. javassist 应用流程与UML类图
4. javassist 特殊语法

javassist 简介

Javassist是一个开源的分析、编辑和创建Java字节码的类库。其主要的优点，在于简单，而且快速。直接使用java编码的形式，而不需要了解虚拟机指令，就能动态改变类的结构，或者动态生成

Javassist 应用场景

监控、动态代理

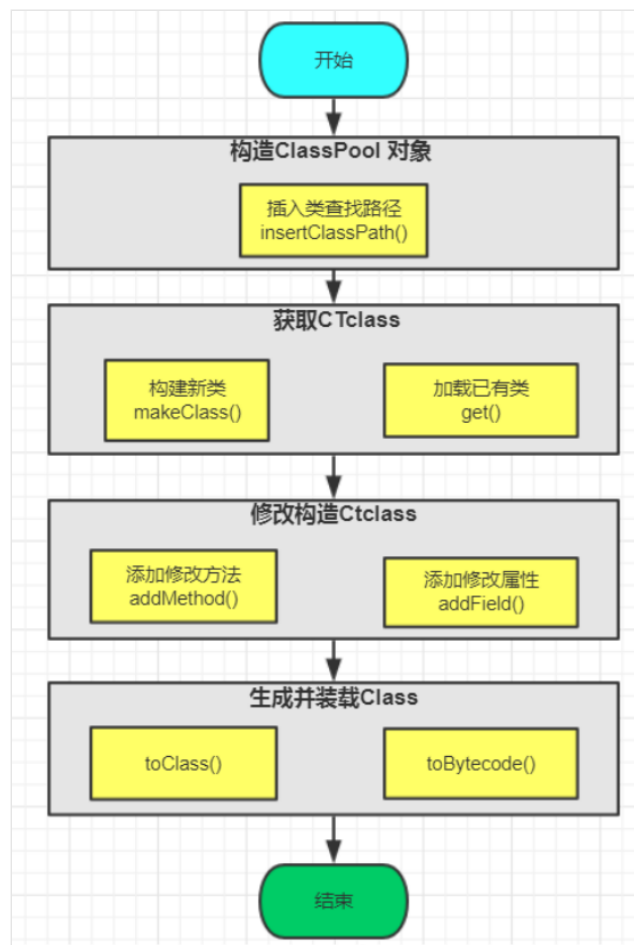
dubbo、myBatis、Spring 都有不同程度的应用。

javassist 使用演示

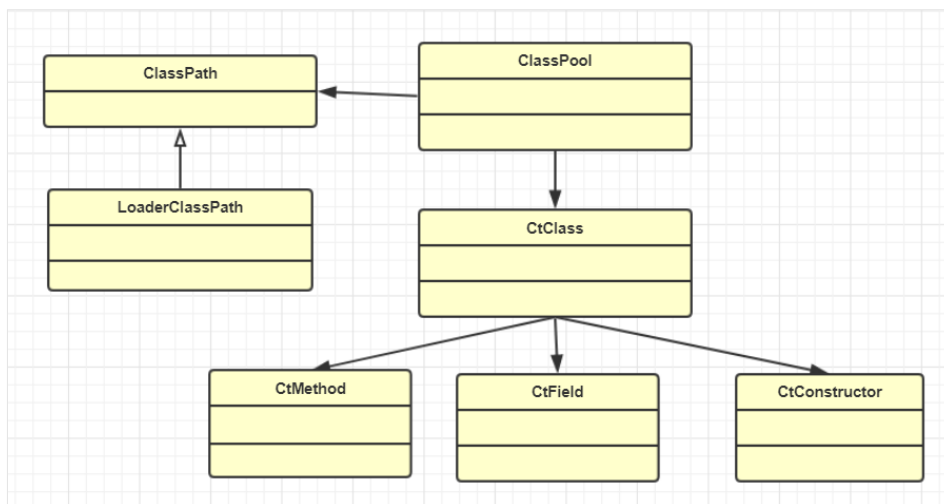
☒ 在运行时修改一个方法代码

javassist API介绍

API执行流程



javassist UML类图



javassist 特殊符号

符号	描述
\$0, \$1, \$2, ...	\$0表示this，其他的表示实际的参数
\$args	参数数组. 相当于new Object[]{ \$1,\$2,...}, 其中的基本类型会被转为包装类型
\$\$	所有的参数，如m(\$\$)相当于m(\$1,\$2...), 如果m无参数则m(\$\$)相当于m()
\$cflow(...)	表示一个指定的递归调用的深度
\$r	用于类型装换，表示返回值的类型.
\$w	将基础类型转换为一个包装类型.如Integer a=(\$w)5; 表示将5转换为Integer。如果不是基本类型则什么都不做。
\$_	返回值，如果方法为void，则返回值为null; 值在方法返回前获得，如果希望发生异常是有返回值（默认值，如nul），需要将insertAfter方法的第二个参数asFinally设置为true
\$sig	方法参数的类型数组，数组的顺序为参数的顺序
\$type	返回类型的class， 如返回Integer则\$type相当于java.lang.Integer.class， 注意其与\$r的区别
\$class	方法所在的类的class

javassist 特殊语法说明：

- a) 不能引用在方法中其它地方定义的局部变量
- b) 不会对类型进行强制检查：如 `int start = System.currentTimeMillis();` 或 `String i="abc";`
- c) 使用特殊的项目语法符号

三、插桩技术应用实践

使用字节码插桩技术监听c3p0数据源。实现类似druid 的Sql 监控

localhost:8080/druid/index.html

Druid Monitor 首页 数据源 SQL监控 SQL防火墙 Web应用 URI监控 Session监控 spring监控 JSON API 重置 记录日志并重置

Stat Index 查看JSON API English | 中文

版本	1.0.24
驱动	com.alibaba.druid.proxy.DruidDriver com.mysql.jdbc.Driver
是否允许重置	false
重置次数	0
java版本	1.7.0_79 重新加载此页
jvm名称	Java HotSpot(TM) 64-Bit Server VM
classpath	/Users/shuaihan/Downloads/apache-tomcat-7.0.67/bin/bootstrap.jar /Users/shuaihan/Downloads/apache-tomcat-7.0.67/bin/tomcat-juli.jar
启动时间	2017-04-18 21:20:25

powered by AlibabaTech & sandzhang & melin & shrek.wang

实现流程：

- ☒ 编写c3p0-agent
- ☒ 通过插桩方式获取 c3p0-DataSource 实例。
 - ☒ `System.getProperties().get("c3p0Source$agent")`
- ☐ 对外提供Http 服务展示DataSource当前状态。

本课程总结

1. javaagent jar包编写与打包
2. javassist 使用流程
3. javassist 特殊语法
4. 字节码插桩实践：监听c3p0数据源。

作业：

继续完善c3p0的监控，并运用到自己实例上去。