1. BUG描述

本质就是一个NoSuchMethodError，如下所示：

java.lang.NoSuchMethodError: org.springframework.core.annotation.AnnotationAwareOrderComparator.sort(Ljava/util/List;)V

at org.springframework.boot.SpringApplication.getSpringFactoriesInstances(SpringApplication.java:394)

at org.springframework.boot.SpringApplication.getSpringFactoriesInstances(SpringApplication.java:383)

at org.springframework.boot.SpringApplication.initialize(SpringApplication.java:249)

at org.springframework.boot.SpringApplication.<init>(SpringApplication.java:225)

at org.springframework.boot.SpringApplication.run(SpringApplication.java:1118)

at org.springframework.boot.SpringApplication.run(SpringApplication.java:1107)

at com.example.demoNoSuchMethodError.DemoNoSuchMethodErrorApplication.main(DemoNoSuchMethodErrorApplication.java:13)

这个异常的意思是AnnotationAwareOrderComparator类缺少sort(Ljava/util/List;)V这个函数，后面的函数使用的是JVM中的函数表示方式。

1. 解决思路
2. 概述

某个包的某个类缺少某个函数，这貌似不是我们的问题，可能是某个包的版本原因导致这个包的这个类没有这个函数，也有可能是这个包的优先级高于同名的其他包，导致其他包被覆盖了，因此我们需要找到这个包的位置。

1. arthas工具

那么，如何找到这个类是从哪个包引入的呢？使用阿里开源的arthas工具来解决，它能够解决以下的问题：

1. 这个类从哪个 jar 包加载的？为什么会报各种类相关的 Exception？
2. 我改的代码为什么没有执行到？难道是我没 commit？分支搞错了？
3. 遇到问题无法在线上 debug，难道只能通过加日志再重新发布吗？
4. 线上遇到某个用户的数据处理有问题，但线上同样无法 debug，线下无法重现！
5. 是否有一个全局视角来查看系统的运行状况？
6. 有什么办法可以监控到JVM的实时运行状态？

通过这个工具来查看这个类是从哪个包引入的，但是首先我们要保证在使用该工具时出现异常的类所在的进程依然在执行，这就需要我们在可能出现异常的代码块中使用try-catch语句块来抓住异常，这里使用Throwable类对象而不是Exception类对象来catch，可能是因为如果使用Exception类的话，由于报错的是一个Error，它无法catch到这个Error，因此会直接报错并且退出，而Throwable类是Error类和Exception类的父类，包含了Error类，因此可以catch这个Error，进入catch代码块中，执行阻塞的语句，保持进程的运行，这样arthas工具可以在进程运行时找出错误。

下载完arthas的jar包后，新建一个文件夹将jar包放进去。之后可以在IDEA的控制台使用，也可以在CMD中使用，进入该文件夹中，输入

java -jar arthas-boot.jar

arthas会扫描JVM启动的进程，我们可以选择对应的进程，进入该进程中，并执行sc命令来查询

sc -d org.springframework.core.annotation.AnnotationAwareOrderComparator

可以看到输出如下所示，spring-2.5.6jar包就被找出来了：

$ sc -d org.springframework.core.annotation.AnnotationAwareOrderComparator

class-info org.springframework.core.annotation.AnnotationAwareOrderComparator

code-source /Users/hengyunabc/.m2/repository/org/springframework/spring/2.5.6.SEC03/spring-2.5.6.SEC03.jar

name org.springframework.core.annotation.AnnotationAwareOrderComparator

isInterface false

isAnnotation false

isEnum false

isAnonymousClass false

isArray false

isLocalClass false

isMemberClass false

isPrimitive false

isSynthetic false

simple-name AnnotationAwareOrderComparator

modifier public

annotation

interfaces

super-class +-org.springframework.core.OrderComparator

+-java.lang.Object

class-loader +-sun.misc.Launcher$AppClassLoader@5c647e05

+-sun.misc.Launcher$ExtClassLoader@689e3d07

classLoaderHash 5c647e05

Affect(row-cnt:1) cost in 41 ms.

使用jad命令查看反编译的AnnotationAwareOrderComparator源码：

$ jad org.springframework.core.annotation.AnnotationAwareOrderComparator

ClassLoader:

+-sun.misc.Launcher$AppClassLoader@5c647e05

+-sun.misc.Launcher$ExtClassLoader@689e3d07

Location:

/Users/hengyunabc/.m2/repository/org/springframework/spring/2.5.6.SEC03/spring-2.5.6.SEC03.jar

/\*

\* Decompiled with CFR 0\_132.

\*/

package org.springframework.core.annotation;

import java.lang.annotation.Annotation;

import org.springframework.core.OrderComparator;

import org.springframework.core.Ordered;

import org.springframework.core.annotation.Order;

public class AnnotationAwareOrderComparator

extends OrderComparator {

protected int getOrder(Object obj) {

Order order;

if (obj instanceof Ordered) {

return ((Ordered)obj).getOrder();

}

if (obj != null && (order = obj.getClass().getAnnotation(Order.class)) != null) {

return order.value();

}

return Integer.MAX\_VALUE;

}

}

Affect(row-cnt:1) cost in 286 ms.

发现，里面确实没有sort函数，终于找到问题所在了，只要将这个jar包排除即可。

1. maven命令

如何找到这个spring的jar包呢？首先，从Maven命令入手，查看maven的依赖树，没有以依赖冲突，可以使用CTRL + F 来搜索jar包。也可以直接使用maven命令搜索指定的包的依赖树，如搜索spring包的依赖树：

mvn dependency:tree -Dincludes=org/springframework/spring

这样过于麻烦，接着找到一个maven插件maven-helper，直接在File -> setting -> Plugins -> 搜索maven helper ，然后点击安装，重启IDEA。

Maven Helper成功后，在左下角会多一个tab页Dependency Analyzer，通过它可以进行如下操作：

1. Conflicts（查看冲突）
2. All Dependencies as List（列表形式查看所有依赖）
3. All Dependencies as Tree（树形式查看所有依赖）

因此，找到了spring包的位置在新添加的一个包中，那么直接在该包的坐标导入中使用以下的方式来排除这个spring依赖：

<!-- Canal客户端-->  
<dependency>  
 <groupId>com.alibaba.otter</groupId>  
 <artifactId>canal.client</artifactId>  
 <version>1.0.24</version>  
 <exclusions>  
 <exclusion>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring</artifactId>  
 </exclusion>  
 </exclusions>  
</dependency>

至此，问题解决！！！