Java8 迁移到 Java11 的过程

作者：Lord-X
原文链接：https://ld246.com/article/1564729155415
来源网站：链滴

□rose□rose　　如果您觉得我的文章对您有帮助的话，记得在GitHub上star一波哈□
ose□rose

□rose□rose　　GitHub_awesome-it-blog □rose□rose

---

0 背景

酷划商业平台容器化之后，应用的启动性能下降明显。经常出现10秒以上的接口响应延迟。分析了应启动时的性能指标，发现CPU负载和使用率同时飙升（甚至打满），后来发现是C2编译器占用大量CP时间片。遂决定优化之。

已经做的优化：

- 调整多个pod的启动顺序：并行 -> 串行
- 调整JIT编译线程数

初步优化后，将延迟降低到了5秒左右。

然后考虑，能不能将每次JIT编译的结果缓存下来，下次应用启动的时候直接使用，减少应用启动初期释执行和收集profile带来的性能影响。

调研发现，目前已有的实现方案有以下几种：

- Oracle Java9以后的AOT（配合Graal编译器使用）
- 阿里JDK-Dragonwell8（基于JDK8，利用其JWarmup缓存JIT编译结果）
- Azul Zing VM的ReadyNow
- IBM J9的AOT

最终决定对前两种进行试验。

PS：本文只讨论OracleJDK的升级过程

1 升级Java11的目的

- 利用AOT解决应用冷启动的性能问题
- 新特性http2的使用
- 容器相关的新特性的使用

2 初期遇到的问题

直接安装Oracle JDK11，将IDE的环境切换到JDK11后，就开始各种报错。例如：

注解PostConstruct找不到了。这个问题是Java11中移除了javax包引起的。

然后开始翻官方文档，有一篇专门针对从JDK8升级到后面新版本的。

传送门：Migrating From JDK 8 to Later JDK Releases

这里面，主要关注哪些会导致应用编译失败或无法启动的更改。

主要有以下这些API移除：

- 移除 java.* APIs 和 javax.* APIs
- 移除 com.sun.*
- 移除某些 jdk.*
- java.awt.peer Not Accessible
- Removed com.sun.image.codec.jpeg Package

另外下面的GC参数组合已失效，如果配置会导致启动报错：

- DefNew + CMS
- ParNew + SerialOld
- Incremental CMS

由于移除了永久代，以下相关配置会导致一些警告：

- -XX:MaxPermSize=size
- -XX:PermSize=size

告警信息如下：

```
Java HotSpot(TM) 64-Bit Server VM warning: Ignoring option MaxPermSize; support was rem
ved in 8.0
```

更详细的信息请参考官方文档的内容。

2.1 解决上面遇到的问题

上文中，切换到JDK11后，javax.annotation.PostConstruct被移除了。我们可以手动将javax.annotation的jar包引入，即可解决。

```
<dependency>
    <groupId>javax.annotation</groupId>
    <artifactId>javax.annotation-api</artifactId>
    <version>1.3.2</version>
</dependency>
```

其他的问题按照同样的思路解决即可。

3 远程debug问题

Java9以后的版本，JPDA的address配置格式发生了一些变化。

Java8及以前，用如下形式：

```
-Xdebug -Xrunjdwp:transport=dt_socket,address=8000,server=y,suspend=n
```

Java9及以后，改为下面这样，否则无法链接到远程端口：

```
# 区别在于address，要变成 *:PORT 格式
-Xdebug -Xrunjdwp:transport=dt_socket,address=*:8000,server=y,suspend=n
```

原因：

可参考StackOverflow的一个issue：What are Java command line options to set to allow JVM to be remotely debugged?

大概是说，Java9及以后的版本默认只支持本地连接到debug，即address=8000，这个格式默认解出来的IP是localhost，改成address=*:8000后，可匹配任何IP。

core-svc/debugger
→ **JDWP socket connector accept only local connections by default**
The JDWP socket connector has been changed to bind to localhost only if no ip address or hostname is specified on the agent command line. A hostname of asterisk (*) may be used to achieve the old behavior which is to bind the JDWP socket connector to all available interfaces; this is not secure and not recommended.

JDK-8041435 (not public)

延伸：

Java5之前，使用-Xdebug和-Xrunjdwp来告知jvm运行在debug模式下，这个方式在以后的版本中然适用，但在此模式下运行的JVM，只会在解释执行的模式下运行，不会有JIT的参与，所以运行较慢因此生产环境一定不要开启这个参数。

后续...

采坑仍在继续，会在GitHub上持续更新。

GitHub_Mr羽墨青衫