1、疑难问题：

用airflow跑程序，程序莫名挂起等待

定位问题：

用jps、jstack -l 进程id查看进程的堆栈，确定是console日志打印问题 - locked <0x00000000e0639f20> (a org.apache.log4j.ConsoleAppender)

分析：可能因为airflow控制台缓存太小，日志写到30条数据就不能在写入了，导致主程序阻塞，Timed\_waiting,

解决办法：去掉log4j的控制台日志输出

2、执行jar

|  |
| --- |
| java -jar -Xms128m -Xmx512m -XX:PermSize=64m -XX:MaxPermSize=128M gmv-1.0-SNAPSHOT-jar-with-dependencies.jar dev |

3、maven将项目（包括依赖的jar）打包称一个jar

|  |
| --- |
| <plugin>  <groupId>org.apache.maven.plugins</groupId>  <artifactId>maven-assembly-plugin</artifactId>  <version>2.5.5</version>  <configuration>  <archive>  <manifest>  <mainClass>gmv.ProductAddToERP</mainClass>  </manifest>  </archive>  <descriptorRefs>  <descriptorRef>jar-with-dependencies</descriptorRef>  </descriptorRefs>  </configuration>  <executions>  <execution>  <id>make-assembly</id>  <phase>package</phase>  <goals>  <goal>single</goal>  </goals>  </execution>  </executions> </plugin> |

注：如果要用java -jar执行jar包，需要指定mainClass（包含main方法的类）

如果要使用其他的main方法启动jar，需要用-cp

|  |
| --- |
| java -Xms128m -Xmx512m -XX:PermSize=64m -XX:MaxPermSize=128M -cp /data1/airflow/myscripts/indonesiaProductAutoShelve/gmv-1.0-SNAPSHOT-jar-with-dependencies.jar gmv.ProductSkuShelfOffSale prod $1 $2 |

|  |
| --- |
| java -jar -Xms128m -Xmx512m /data1/airflow/myscripts/indonesiaProductAutoShelve/gmv-1.0-SNAPSHOT-jar-with-dependencies.jar prod yinni $1 $2 |

4、分模块的maven项目，上传jar，必须要将父pom.xml文件一同上传的maven仓库（否则会下载不下来）

5、不同数据源怎么使用事物来保证数据的一致性？

6、问题：加载了文件，在java类中还是不能通过@Value注解注入值

<context:property-placeholder location="classpath:jollychic.properties" ignore-unresolvable="true"/>

在spring的xml配置文件中当有多个\*.properties文件需要加载时

解决：添加ignore-unresolvable="true"

7、dubbo的@Refference注解不能使用占位符，例如@Reference(version = "${srevice.version}")

这样读取不到service.version的值。

一般服务的版本号在统一配置在配置文件中，需要用到占位符，

解决办法：用xml配置，在xml配置中可以使用占位符

8、eclipse查看类的继承关系 选中要查看的类，按F4

9、java.net.SocketException: Connection reset by peer

**第4个异常是java.net.SocketException:（Connection reset或者Connectreset by peer:Socket write error）。**

**原因 ：用户在发送一个请求后，再次发送另一个请求，导致第一个请求中断或者**

①：服务器的并发连接数超过了其承载量，服务器会将其中一些连接Down掉；   
②：客户关掉了浏览器，而服务器还在给客户端发送数据；   
③：浏览器端按了Stop

该异常在客户端和服务器端均有可能发生，引起该异常的原因有两个，第一个就是如果一端的Socket被关闭（或主动关闭或者因为异常退出而引起的关闭），另一端仍发送数据，发送的第一个数据包引发该异常(Connect reset by peer)。另一个是一端退出，但退出时并未关闭该连接，另一端如果在从连接中读数据则抛出该异常（Connection reset）。简单的说就是在连接断开后的读和写操作引起的。

解决办法：