在调试spring应用时，动辄几十秒，甚至有的应用上分钟的启动速度，会让整个调试速度慢下来了。等待时间让人抓狂。不知道大家是如何加速spring应用调试速度的，在此分享下我的一次加速过程。欢迎补充指正。

**环境**

**配置：**

1. thinkpad t410
2. 内存：4G内存
3. CPU：Intel P8700 双核2.53GHZ
4. 系统：WIN XP
5. 开发工具：Intellij IDEA 12.0.4
6. Maven + spring3.2.3 + hibernate4.2.2+Spring data jpa 1.3.1

**未优化前spring容器启动速度：**

16890毫秒 =（14609毫秒（ContextLoaderListener加载的）+2281毫秒（Springmvc加载的）

**优化后spring容器启动速度：**

7797毫秒 =（6563毫秒（ContextLoaderListener加载的）+1234毫秒（Springmvc加载的）

速度提升了一半多，而且以后在调试阶段，大部分就停留在这个时间左右。

注意：此处只是spring容器启动速度，不包括服务器启动时的速度。因为我的系统好久没清理了，否则可能速度会更快。

**加速Spring**

**1、扫描注解Bean**

写比较精确的扫描路径，如扫描@Service和@Repository：

1. **<context:component-scan** base-package="com.sishuok.es.\*\*.repository,com.sishuok.es.\*\*.service,com.sishuok.es.\*\*.extra"**>**

这样写，比直接写com.sishuok.es速度要快很多，因为这样扫描的class会很少。

还有，如springmvc 扫描：

1. **<context:component-scan** base-package="com.sishuok.es.\*\*.web.controller" use-default-filters="false"**>**

此处只扫描项目的web.controller包，这样扫描的class也很少。

还有如事务的扫描：

1. execution(\* com.sishuok.es..service..\*+.\*(..)

还有如使用spring data jpa时也是这样：

1. **<jpa:repositories**
2. base-package="com.sishuok.es.\*\*.repository"

这里需要大家有良好的分包，否则无法优化。

**2、延迟加载你的bean**

常见的方式是在配置文件中在<beans>上加：

1. default-lazy-init="true"

2.1、这种方式只对xml声明的bean有效；

2.2、注解扫描的bean无效，如@Service，需要使用@Lazy指定，但这样太麻烦，需要一个一个的配置；

2.3、 还有就是如果你使用springmvc，lazy-init几乎没啥用，因为springmvc容器在启动时会通过 DefaultAnnotationHandlerMapping查找相关的带有@RequestMapping的bean并注册请求映射；所以相关的如 Service/Repository也级联非lazy-init；

因此我写了个工具：SpeedUpSpringProcessor，其作用是：lazy-init所有bean，包括注解的bean；对于【2.3】后续介绍解决方案；具体配置请参考最后。

**3、移除调试阶段不相干的bean**

有些bean在调试阶段我们并不需要，如我们在测试用户模块时，可能不需要测试权限模块；此时我们可以把不相干的bean移除掉；具体配置请参考最后。

这样的话，可以考虑如把@Controller的bean移除，这样的话如Service/Repository就可以lazy-init了。

常见的可以移除的如：

* 任务调度器（quartz）、AOP相关等等；
* 此处需要合理的分包，否则无法应用或应用困难。

**4、删除无用属性**

如在测试shiro时，可能不需要remember的功能，此时可以把属性移除/禁用（即将值设置为false）；具体配置请参考最后。

**5、替换正式机数据源为最快的数据源**

如此处我把DruidDataSource数据源直接替换为org.springframework.jdbc.datasource.DriverManagerDataSource，这个速度最快；

**6、替换jackson为fastjson**

此处测试了下jackson速度比fastjson慢许多的。支持国产。

**7、项目分模块开发**

如果项目模块比较多，可以考虑放弃注解，而使用xml配置方式+约定。因为实际做项目时可能把配置分到多个配置文件，此时我尝试了下合并到一个，几乎没啥速度提升，所以还是分开存好。

到此spring容器启动速度算是比较快了，不知道大家还有没有好的策略。欢迎指点。

**加速Hibernate/JPA**

此处以org.springframework.orm.jpa.LocalContainerEntityManagerFactoryBean为例。

**1、 精确化packagesToScan**

和之前的spring一样，写比较精确的实体扫描路径

1. **<property** name="packagesToScan" value="com.sishuok.es.\*\*.entity"**/>**

**2、generateDdl=false 禁用掉**

没必要每次都生成ddl

**3、 禁用JSR-303验证**

默认情况下是AUTO，会根据classpath下是否有jsr-303实现来自动注册；

1. <!-- 使用自定义的validator进行jsr303验证 -->
2. **<entry** key="javax.persistence.validation.factory" value-ref="validator"**/>**
3. <!-- jsr303验证模式 因为其要么验证 要么不验证 不能按照规则走 所以此处禁用 -->
4. <!-- #http://docs.jboss.org/hibernate/entitymanager/3.6/reference/en/html/configuration.html -->
5. **<entry** key="javax.persistence.validation.mode" value="NONE"**/>**

此处validator 直接引用我们项目中定义的，而不是让hibernate再去new一个。而且也不推荐在这验证，具体为什么，请参考我的《我是这样认识注解和XML的》。

**4、如果你的项目都是注解，此时就没必要扫描hbm了，禁用掉**

1. **<entry** key="hibernate.archive.autodetection" value="class"**/>**

**5、如果你不使用NamedQuery，禁用掉**

1. **<entry** key="hibernate.query.startup\_check" value="false"**/>**

**6、在调试阶段禁用掉二级缓存**

通过如上手段，我的spring容器启动速度提升了一半多。大家还有好的优化策略吗？如果有欢迎补充。具体配置请参考最后。

**之前提到的SpeedUpSpringProcessor配置**

1. <!-- 优化spring启动 通过移除bean定义 和 lazy init 实现 -->
2. **<bean** class="com.sishuok.es.common.spring.SpeedUpSpringProcessor"**>**
3. <!-- 需要从bean定义中移除的bean的名字 -->
4. **<property** name="removedBeanNames"**>**
5. **<list>**
6. <!-- spring-config-quartz.xml -->
7. **<value>**scheduler**</value>**
8. **<value>**autoClearDeletedRelationTrigger**</value>**
9. **<value>**autoClearExpiredOrDeletedmMessageTrigger**</value>**
10. **<value>**autoClearDeletedRelationJob**</value>**
11. **<value>**autoClearExpiredOrDeletedmMessageJob**</value>**
13. <!-- spring-config-shiro.xml -->
14. **<value>**rememberMeCookie**</value>**
15. **<value>**rememberMeManager**</value>**
16. **<value>**shiroCacheManager**</value>**
17. **<value>**sessionValidationScheduler**</value>**
18. **<value>**sessionValidationScheduler**</value>**
19. <!-- spring-mvc.xml -->
20. **<value>**multipartResolver**</value>**
22. <!-- spring-config-monitor.xml -->
23. **<value>**druidStatInterceptor**</value>**
24. **<value>**druidAdvisor**</value>**
25. **</list>**
26. **</property>**
27. <!-- 需要从bean定义中移除的bean的属性 -->
28. <!--替换掉的属性值 see removedBeanProperties 只支持简单属性-->
29. **<property** name="removeOrReplaceBeanProperties"**>**
30. **<list>**
31. <!-- spring-config-shiro.xml -->
32. **<value>**sessionManager#cacheManager**</value>**
33. **<value>**sessionManager#cacheManager**</value>**
34. **<value>**sessionManager#sessionValidationScheduler**</value>**
35. **<value>**securityManager#rememberMeManager**</value>**
37. <!-- spring-config.xml -->
38. **<value>**entityManagerFactory#jpaPropertyMap#hibernate.default\_batch\_fetch\_size"**</value>**
39. **<value>**entityManagerFactory#jpaPropertyMap#hibernate.max\_fetch\_depth"**</value>**
40. **<value>**entityManagerFactory#jpaPropertyMap#hibernate.generate\_statistics**</value>**
41. **<value>**entityManagerFactory#jpaPropertyMap#hibernate.bytecode.use\_reflection\_optimizer**</value>**
42. **<value>**entityManagerFactory#jpaPropertyMap#hibernate.cache.use\_second\_level\_cache=false**</value>**
43. **<value>**entityManagerFactory#jpaPropertyMap#hibernate.cache.use\_query\_cache**</value>**
44. **<value>**entityManagerFactory#jpaPropertyMap#hibernate.cache.region.factory\_class**</value>**
45. **<value>**entityManagerFactory#jpaPropertyMap#hibernate.cache.use\_structured\_entries**</value>**
46. **<value>**entityManagerFactory#jpaPropertyMap#net.sf.ehcache.configurationResourceName**</value>**
47. **</list>**
48. **</property>**
50. <!-- 需要从bean定义中移除指定的类类型 正则表达式-->
51. **<property** name="removedClassPatterns"**>**
52. **<list>**
53. **<value>**com\.sishuok\.es\.showcase.\***</value>**
54. **<value>**com\.sishuok\.es\.monitor.\***</value>**
55. **<value>**com\.sishuok\.es\.extra\.aop.\***</value>**
56. **<value>**com\.sishuok\.es\.extra\.quartz.\***</value>**
57. **<value>**com\.sishuok\.es\.conf.\***</value>**
58. <!--<value>com\.sishuok\.es\.personal.\*\.web\.controller.\*</value>-->
59. <!--<value>com\.sishuok\.es\.sys.\*\.web\.controller.\*</value>-->
60. **</list>**
61. **</property>**
63. <!-- 指定非延迟加载的bean-->
64. **<property** name="noneLazyBeanNames"**>**
65. **<list>**
66. **<value>**domainClassConverter**</value>**
67. **</list>**
68. **</property>**
69. **</bean>**
70. 默认所有bean lazy-init；
71. removedClassPatterns：正则表达式，即可以移除的bean的class路径模式，bean class匹配该模式的将移除；此处需要良好的分包，否则不好应用；
72. removedBeanNames：即在调试期间可以移除的bean；
73. removeOrReplaceBeanProperties： 调试期间可以删除/替换掉的bean属性；如移除shiro的sessionManager的cacheManager；如禁用hibernate二级缓 存：entityManagerFactory#jpaPropertyMap#hibernate.cache.use\_second\_level\_cache=false
74. noneLazyBeanNames：有些bean不能lazy-init；排除掉。

**具体实现请参考：**

可以直接下载使用。

SpeedUpSpringProcessor：

https://github.com/zhangkaitao/es/blob/master/common/src/main/java/com/sishuok/es/common/spring/SpeedUpSpringProcessor.java

spring-speed-up.xml：

https://github.com/zhangkaitao/es/blob/master/web/src/main/resources/spring-speed-up.xml

其他提到的配置文件都在：

https://github.com/zhangkaitao/es/tree/master/web/src/main/resources

**开启/关闭：**

此处我使用了spring的profile：

1. **<beans** profile="development" **>**

即只有当System.getProperties中有spring.profiles.active=developement才执行调试模式，所以如果没有该配置还是走的正常流程，对系统没有影响，所以此处大家可以使用：

1、jetty内嵌执行时设置该属性

1. **<plugin>**
2. **<groupId>**org.mortbay.jetty**</groupId>**
3. **<artifactId>**jetty-maven-plugin**</artifactId>**
4. **<version>**${jetty.version}**</version>**
5. **<configuration>**
6. ---省略
7. <!-- spring profile  -->
8. **<systemProperties>**
9. **<systemProperty>**
10. **<name>**spring.profiles.active**</name>**
11. **<value>**development**</value>**
12. **</systemProperty>**
13. **</systemProperties>**
14. **</configuration>**
15. **</plugin>**

2、写多个bat文件分别执行不同的情况。