现象：

应用程序收到监控报警，报警信息描述是频繁的fullgc

排查：

频繁的fullgc第一时间想到的是老年代内存不够用了，通过jmap –heap发现老年代内存使用率99%（如图1），既然老年代内存使用率99%，而且GC回收不了，可能会不断的创建对象，想看哈对象的内存分配情况，于是执行命令jmap –histo （见图2）发现ConcurrentHashMap大对象，占用大概2G的内存，为什么ConcurrentHashMap这么大，于是执行jamp –dump利用mat进行分析，分析如图3

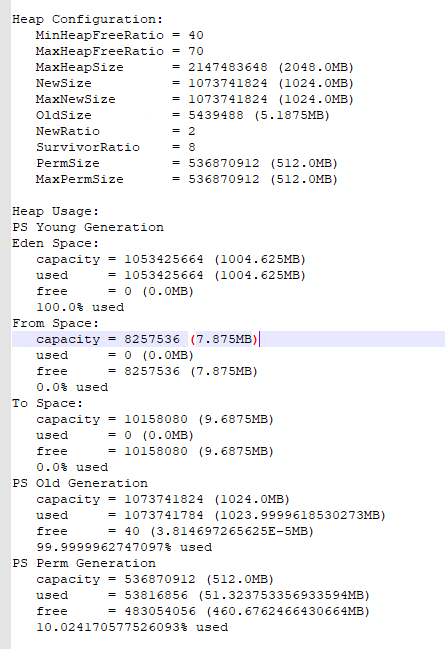


图1

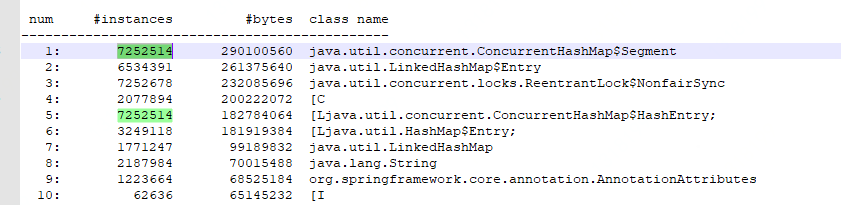


图2

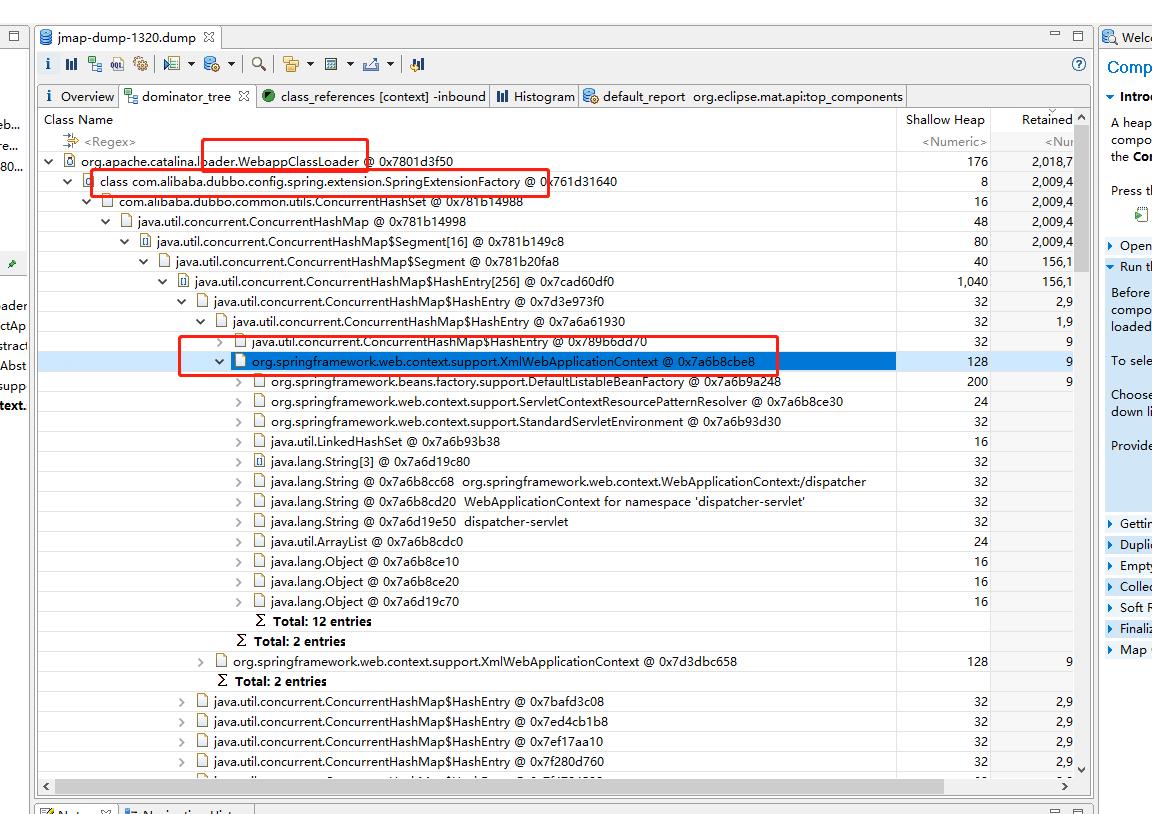


图3

通过MAT分析大对象树发现ConcurrentHashMap里面存储的是XmlWebApplicationContext，为什么会存储这么多上下文对象，而且这个Map的所有者是Dubbo，于是查看SpringExtensionFactory源码，具体如图4，

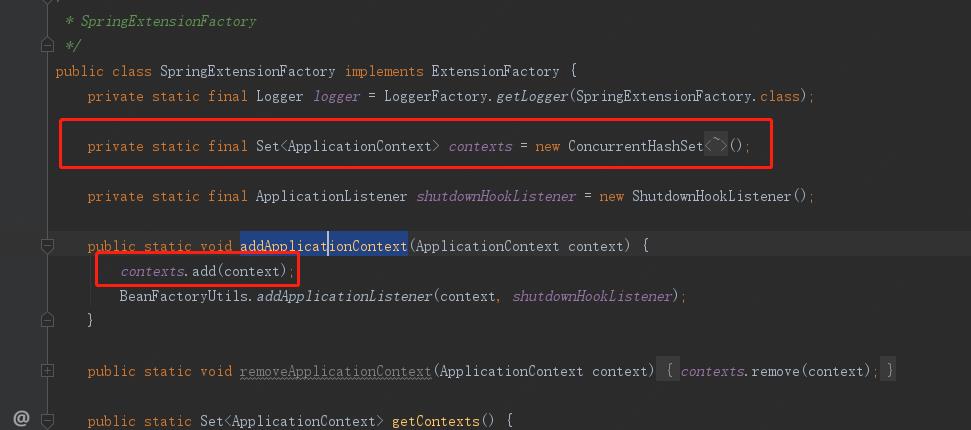


图4

那什么情况下会触发添加这个上下文呢，于是查找调用SpringExtensionFactory的地方如图5、图6

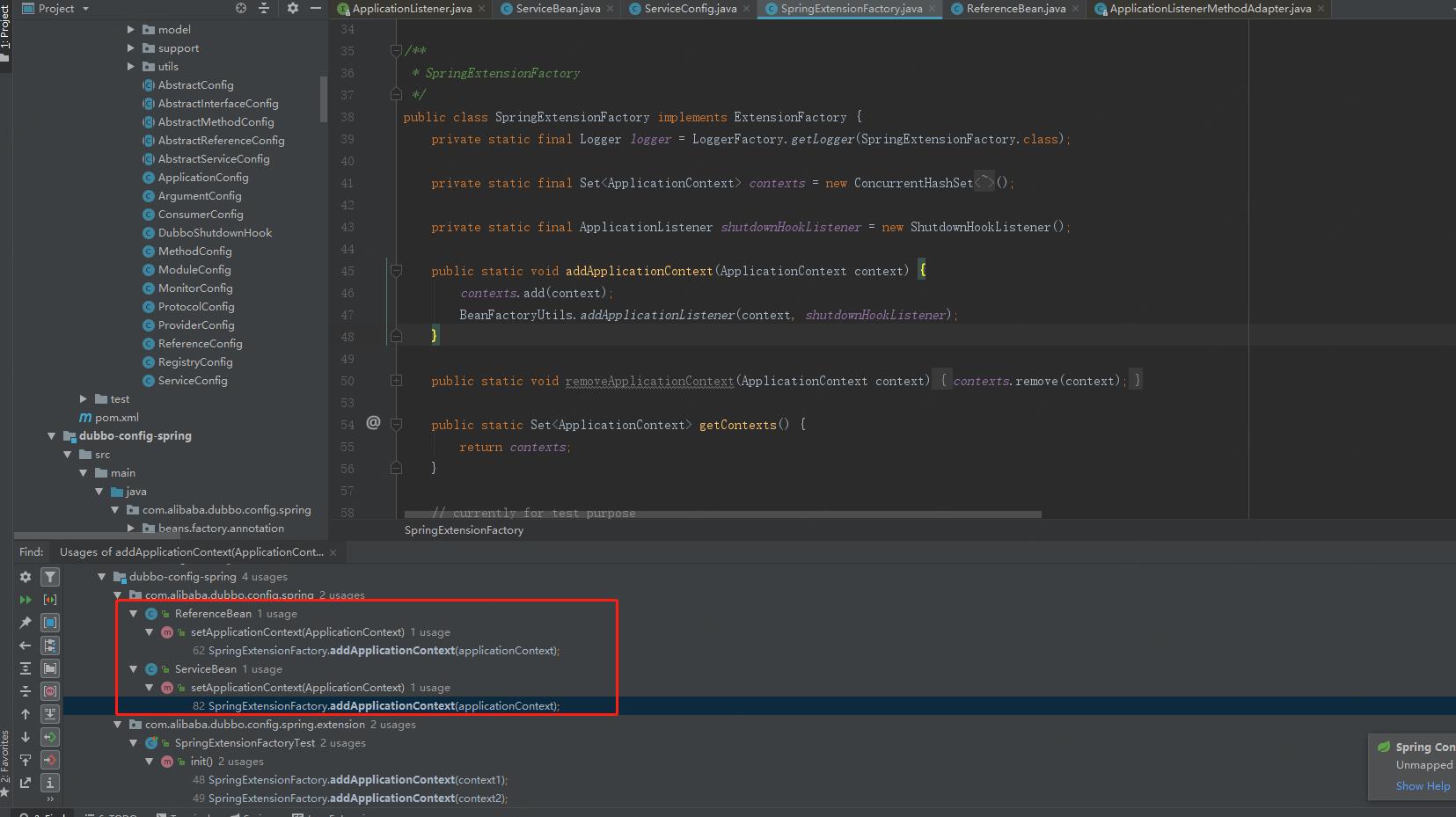


图5

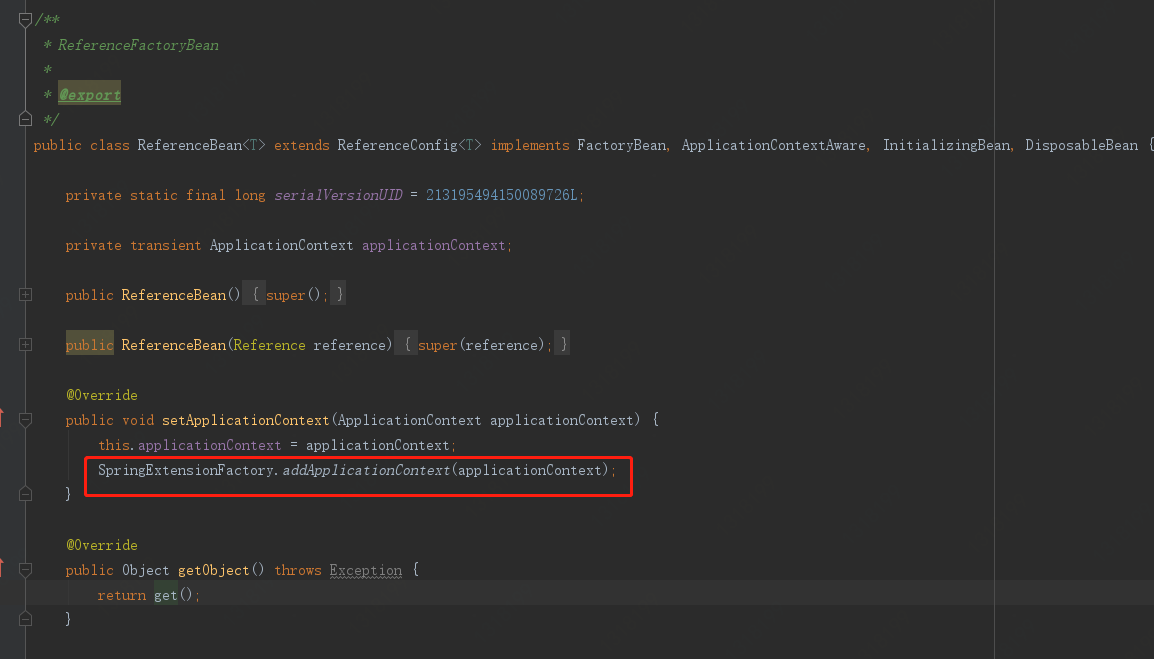


图6

虽然找到了什么地方会去添加Context,但是为什么Dubbo会不断的放入Context，按道理只有一个Context，怎么会不断的创建Context呢，通过查找日志发现ZKClint版本过高，JDK不符合，Duubo初始化消费者时通过ZK去找服务，但是ZKClient class版本不对于是整个Spring容器初始化失败了，具体信息如图7

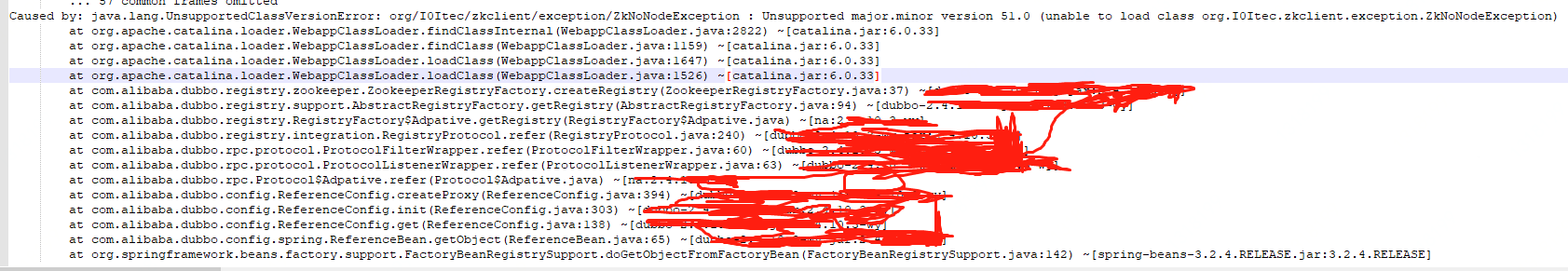


图7

虽然Spring容器初始化失败，但是为什么会不断的初始化容器呢，进一步分析日志发现日志中大量出现ERROR DispatcherServlet - Context initialization failed，为什么会大量出现DispatcherServlet初始化容器的异常呢，肯定有大量外部请求造成的，本身应用没有这么大的请求量，请求量从哪里来的呢，进一步分析得知运维有心跳检测，所以到此本次OOM原因以及清楚了，于是通过模拟让Spring容器初始化失败，然后发送请求，通过VM监控可以明显看到ConcurrentHashMap在增长，但是想不明白的是为什么Tomcat明确感知到Web容器初始化失败了，还要启动成功了。