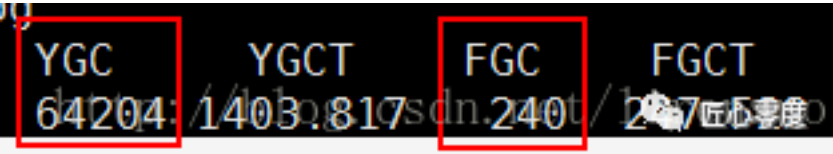


JVM菜鸟进阶高手之路一

2017-09-09 匠心零度

今天在JVMPocket群里面看见，阿牛发了一个gc截图，之后ak47截图了



说特别恐怖，我就觉得好奇，去看看服务情况，截图日志如下

jstat -gcutil 3s 10										
S0	S1	E	O	P	YGC	YGCT	FGC	FGCT	GCT	
0.00	0.00	28.17	92.18	1.09	2	1.611	125228	10477.732	10479.343	
0.00	0.00	57.25	92.18	1.09	2	1.611	125228	10477.732	10479.343	
0.00	0.00	74.72	92.18	1.09	2	1.611	125228	10477.732	10479.343	
0.00	0.00	96.65	92.18	1.09	2	1.611	125228	10477.732	10479.343	
0.00	0.00	44.58	92.18	1.09	2	1.611	125229	10477.816	10479.427	
0.00	0.00	53.59	92.18	1.09	2	1.611	125229	10477.816	10479.427	
0.00	0.00	84.22	92.18	1.09	2	1.611	125229	10477.816	10479.427	
0.00	0.00	13.60	92.18	1.09	2	1.611	125230	10477.898	10479.509	
0.00	0.00	34.10	92.18	1.09	2	1.611	125230	10477.898	10479.509	
0.00	0.00	60.20	92.18	1.09	2	1.611	125230	10477.898	10479.509	

关于jstat命令详情可以参考：<https://docs.oracle.com/javase/8/docs/technotes/tools/unix/jstat.html>

高手就是高手，就通过这个，直接提出1，就是达到old的92%的阈值了 不断做cms gc 2.估计是不断做system gc 这些都是猜测，让执行

jstat -gccause pid 看看情况，为什么说执行gccause呢？

尤其ygc没有，只有fgc，那gccause出来的一定就是我们想要的

如果有ygc，这个命令就不好用了，很容易吧我们想要的东西覆盖

看看jstat的说明

Jstat是JDK自带的一个轻量级小工具。全称“Java Virtual Machine statistics monitoring tool”，它位于Java的bin目录下，主要利用JVM内建的指令对Java应用程序的资源 and 性能进行实时的命令行的监控，包括了对Heap size和垃圾回收状况的监控。可见，Jstat是轻量级的、专门针对JVM的工具，非常适用。

jstat工具特别强大，有众多的可选项，详细查看堆内各个部分的使用量，以及加载类的数量。使用时，需加上查看进程的进程id，和所选参数。参考格式如下：

jstat -options

可以列出当前JVM版本支持的选项，常见的有

- l class (类加载器)
- l compiler (JIT)
- l gc (GC堆状态)
- l gccapacity (各区大小)
- l gccause (最近一次GC统计和原因)
- l gcnew (新区统计)
- l gcnewcapacity (新区大小)

- 1 gcold (老区统计)
- 1 gcoldcapacity (老区大小)
- 1 gcpermcapacity (永久区大小)
- 1 gcutil (GC统计汇总)
- 1 printcompilation (HotSpot编译统计)



通过这个就排除了是执行system gc

在通过

jstat -gc pid 查看gc堆状态



看到这里大家应该都看出问题了，我靠什么情况，old512K

把jvm参数贴出：

-Xms2048m -Xmx5120m -XX:PermSize=4096M -XX:-HeapDumpOnOutOfMemoryError -XX:MaxNewSize=5120m -XX:MaxPermSize=3072m

-Xmx5120m MaxNewSize=5120m old就没有空间了

修改参数

-Xms5120m -Xmx5120m -XX:PermSize=128M -XX:MaxPermSize=512M -XX:-HeapDumpOnOutOfMemoryError -Xmn1512M

再观察情况，

jstat -gcutil pid 3s 30，看一下90s内ygc次数和ygct的时间变化



完美，一次0.005，才5ms

有个参数可以把ygc耗时花在哪里打出来

刚刚说的那个参数是：-XX:+PrintGCApplicationStoppedTime

该参数如何参考，查看呢？ 在微信小程序里面搜索：JVMPocket，这个小程序是笨神大大提供的
结果如下：



这几个参数建议也加上去

```
-XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/data/log/gclog/ -Xloggc:/data/log/gclog/gc.log -  
XX:+UseConcMarkSweepGC -XX:+UseParNewGC -XX:CMSInitiatingOccupancyFraction=75 -  
XX:+UseCMSInitiatingOccupancyOnly
```

显示申明cms+ParNew，设定old区75%时就回收

但是应用启动前需要提前创建目录/data/log/gclog/

参数不明白啥意思 去搜索微信小程序

最后感谢笨神，感谢阿飞！