

图文 051、动手实验：使用jmap和jhat摸清线上系统的对象分布

1574 人次阅读 2019-08-20 07:00:00

详情 评论

动手实验：

使用jmap和jhat摸清线上系统的对象分布

狸猫技术窝专栏上新，基于**真实订单系统**的消息中间件（mq）实战，重磅推荐：



相关频道



从0开始
战高手
已更新1

未来3个月，我的好朋友原子弹大侠将带你一起，全程实战，360度死磕MQ

(点击下方蓝字进行试听)

[从0开始带你成为消息中间件实战高手](#)

重要说明：

如何提问：每篇文章都有评论区，大家可以尽情在评论区留言提问，我都会逐一答疑

(ps：评论区还精选了一些小伙伴对**专栏每日思考题**的作答，有的答案真的非常好！大家可以通过看别人的思路，启发一下自己，从而加深理解)

如何加群：购买了狸猫技术窝专栏的小伙伴都可以加入**狸猫技术交流群**。

(群里有不少**一二线互联网大厂**的助教，大家可以一起讨论交流各种技术)

具体**加群方式**请参见文末。

(注：以前通过其他专栏加过群的同学就不要重复加了)



狸猫技术窝

进店逛

1、前文总结

上一篇文章我们给大家介绍了一个平时工作中非常实用的工具，jstat。

用jstat就可以非常轻松便捷的了解线上系统的运行状况，从新对象增速、Young GC触发频率以及耗时，再到对象进入老年代的增速以及Full GC触发频率以及耗时，可以完全摸清楚线上系统的JVM运行情况，为可能要做的优化做准备。

本文我们继续给大家介绍两个平时工作里非常实用的工具，jmap和jhat。

这两个工具可以帮助我们观察线上JVM中的对象分布，了解到你的系统平时运行过程中，到底哪些对象占据了主角位置，他们占据了多少内存空间，让你对你的系统运行有更加细致的了解。

2、使用jmap了解系统运行时的内存区域

其实如果单单只是要了解JVM的运行状况，然后去进行JVM GC优化，通常来说jstat就完全够用了

但是有的时候可能我们会发现JVM新增对象的速度很快，然后就想要去看看，到底什么对象占据了那么多的内存。

如果发现有的对象在代码中可以优化一下创建的时机，避免那种对象对内存占用过大，那么也许甚至可以去反过来优化一下代码。

当然，其实如果不是出现OOM那种极端情况，也并没有那么大的必要去着急优化代码。

但是这篇文章我们来学习一下如何了解线上系统jvm中的对象分布，也是有好处的，比如之前我们在上周的案例中就发现年轻代里总是有500kb左右的未知对象，大家是不是会很好奇？如果可以看到jvm中这500kb的对象到底是什么就好了，所以学习一下这个技巧是有用的。

先看一个命令：**jmap -heap PID**

这个命令可以打印出来一系列的信息，我们就不长篇大论的粘贴出来具体的信息了，因为内容篇幅太大了，其实也没太大意义，因为里面的东西大家自己看字面意思都能看懂的。我们就简单给大家说一下这里会打印出来什么东西。

大致来说，这个信息会打印出来堆内存相关的一些参数设置，然后就是当前堆内存里的一些基本各个区域的情况

比如Eden区总容量、已经使用的容量、剩余的空间容量，两个Survivor区的总容量、已经使用的容量和剩余的空间容量，老年代的总容量、已经使用的容量和剩余的容量。

但是这些信息大家会想了，其实jstat已经有了啊！对的，所以一般不会用jmap去看这些信息，毕竟他信息还没jstat全呢，因为没有gc相关的统计。

3、使用jmap了解系统运行时的对象分布

其实jmap命令比较有用的一个使用方式，是如下的：

jmap -histo PID

这个命令会打印出来类似下面的信息：

| num | #instances | #bytes | class name |
|-----|------------|---------|--|
| 1: | 46608 | 1111232 | java.lang.String |
| 2: | 6919 | 734516 | java.lang.Class |
| 3: | 4787 | 536164 | java.net.SocksSocketImpl |
| 4: | 15935 | 497100 | java.util.concurrent.ConcurrentHashMap\$Node |
| 5: | 28561 | 436016 | java.lang.Object |

这个就很有意思了，各位同学看上述打印出来的东西，他会按照各种对象占用内存空间的大小降序排列，把占用内存最多的对象放在最上面。

所以如果你只是想要简单的了解一下当前jvm中的对象对内存占用的情况，只要直接用jmap -histo命令即可，非常好用

你可以快速了解到当前内存里到底是哪个对象占用了大量的内存空间。

4、使用jmap生成堆内存转储快照

但是如果你仅仅只是看一个大概，感觉就只是看看上述那些对象占用内存的情况，感觉还不够，想要来点深入而且仔细点的

那就可以用jmap命令生成一个堆内存快照放到一个文件里去，用如下的命令即可：

jmap -dump:live,format=b,file=dump.hprof PID

这个命令会在当前目录下生成一个dump.hprof文件，这里是二进制的格式，你不能直接打开看的，他把这一时刻JVM堆内存里所有对象的快照放到文件里去了，供你后续去分析。

5、使用jhat在浏览器中分析堆转出快照

接着就可以使用jhat去分析堆快照了，jhat内置了web服务器，他会支持你通过浏览器来以图形化的方式分析堆转储快照

使用如下命令即可启动jhat服务器，还可以指定自己想要的http端口号，默认是7000端口号：

```
jhat dump.hprof -port 7000
```

接着你就在浏览器上访问当前这台机器的7000端口号，就可以通过图形化的方式去分析堆内存里的对象分布情况了。

这里我们先简单介绍一下，后面马上有两个案例，jstat、jmap、jhat我们都会带着大家动手来玩一下的。

6、今日思考题

今天留给大家的作业，就是自己去线上系统的机器上，用jmap -histo看看对象大致分布情况，然后用jmap生成一个堆转储快照，再用jhat分析一下堆转储快照，看看当前系统运行的时候，各种对象的分布情况。

大家可以到评论区里说说自己看堆转储快照的感受，是不是发现自己越来越了解JVM是如何运行的了呢？

End

专栏版权归公众号狸猫技术窝所有

未经许可不得传播，如有侵权将追究法律责任

如何加群？

添加微信号：Lvgu0715，（微信名：绿小九），狸猫技术窝管理员

发送 Jvm专栏的购买截图

由于是人工操作，发送截图后请耐心等待被拉群

最后再次提醒：通过其他专栏加过群的同学，就不要重复加了

狸猫技术窝其他精品专栏推荐：

[21天互联网java进阶面试训练营（分布式篇）](#)