

[JVM]OOM实例分析



伤口不该结疤

关注

赞赏支持

[JVM]OOM实例分析



伤口不该结疤 关注

1 2017.02.06 16:55:10 字数 1,031 阅读 2,671

• 深入理解Java虚拟机

- [\[JVM\]Java内存区域与垃圾收集 - 思维导图](#)
- [\[JVM\]类加载机制 - 思维导图](#)
- [\[JVM\]OOM实例分析](#)
- [\[JVM\]理解Class文件\(1\): 手动解析常量池](#)
- [\[JVM\]理解GC日志](#)
- [\[JVM\]理解Class文件\(2\)](#)

• 引言

本文内容总结自《深入理解Java虚拟机：JVM高级特性与最佳实践（第2版）》中的第2章。

- 通过设置**虚拟机的启动参数**来模拟内存区域溢出的情况
- 通过异常信息判断出是哪个内存溢出了，知道什么样的代码可能会导致内存溢出

推荐阅读

最详细的JVM&GC讲解

阅读 10,215

JVM 内存模型 Stack Heap 文章选摘

阅读 1,824

Android内存管理分析总结

阅读 7,726

jQuery实现轮播

阅读 181

猎魔人

阅读 100

写下你的评论...



评论0



赞56



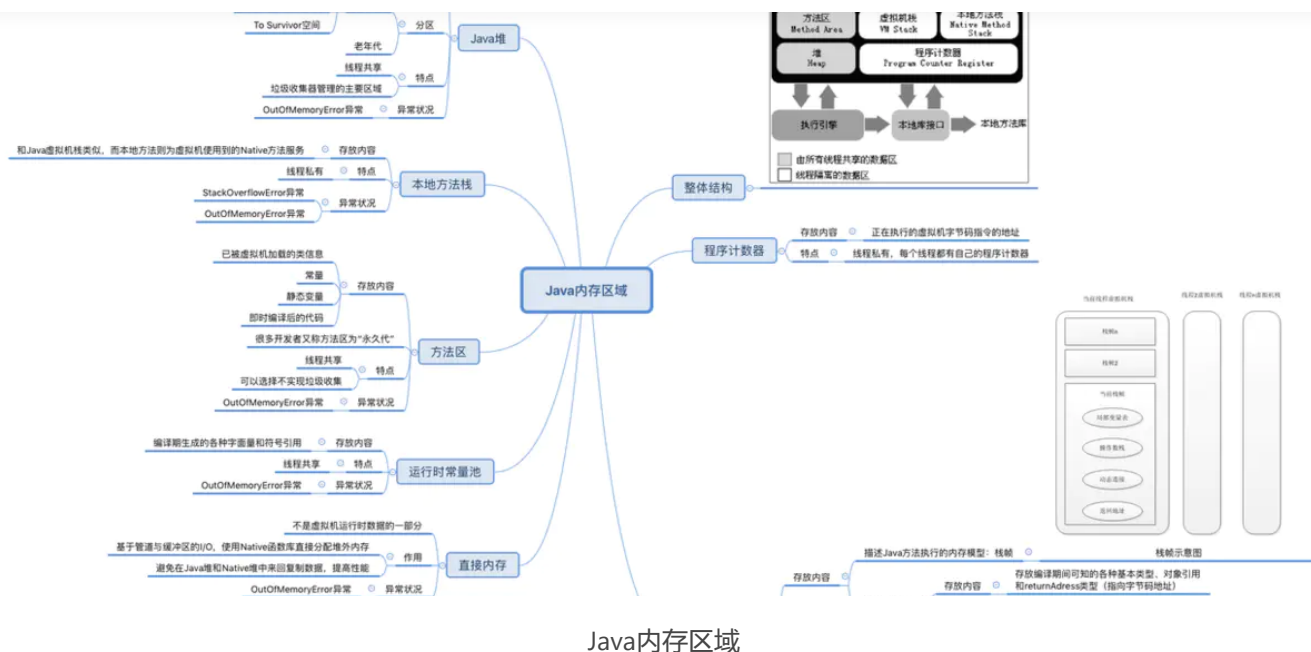
[JVM]OOM实例分析



伤口不该结疤

关注

赞赏支持



推荐阅读

[最详细的JVM&GC讲解](#)

阅读 10,215

[JVM 内存模型 Stack Heap 文章选摘](#)

阅读 1,824

[Android内存管理分析总结](#)

阅读 7,726

[jQuery实现轮播](#)

阅读 181

[猎魔人](#)

阅读 100

• 设置虚拟机的启动参数

在使用代码进行验证之前，我们需要先了解如何设置虚拟机的启动参数，设置方法有两种。

• 使用命令行执行程序时

使用命令行运行程序时，直接在java后面跟上参数和程序名称

```
1 | D:\>javac HeapOOM.java
2 |
3 | D:\>java HeapOOM
```

写下你的评论...



评论0



赞56



[JVM]OOM实例分析



伤口不该结疤

关注

赞赏支持

```
Exception in thread "main" java.lang.OutOfMemoryError: Java heap space
10      at java.util.Arrays.copyOf(Arrays.java:2245)
11      at java.util.Arrays.copyOf(Arrays.java:2219)
12      at java.util.ArrayList.grow(ArrayList.java:242)
13      at java.util.ArrayList.ensureExplicitCapacity(ArrayList.java:216)
14      at java.util.ArrayList.ensureCapacityInternal(ArrayList.java:208)
15      at java.util.ArrayList.add(ArrayList.java:440)
16      at HeapOOM.main(HeapOOM.java:16)
```

推荐阅读

[最详细的JVM&GC讲解](#)

阅读 10,215

[JVM 内存模型 Stack Heap 文章选摘](#)

阅读 1,824

[Android内存管理分析总结](#)

阅读 7,726

[jQuery实现轮播](#)

阅读 181

[猎魔人](#)

阅读 100

• 使用IDE执行程序时

直接在VM Arguments中设置

写下你的评论...



评论0



赞56



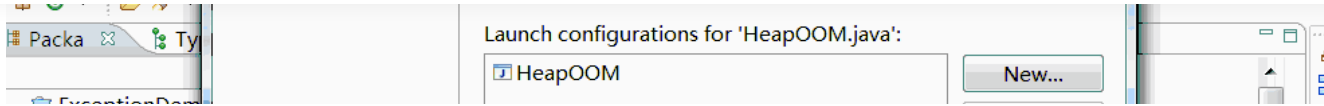
[JVM]OOM实例分析



伤口不该结疤

关注

赞赏支持



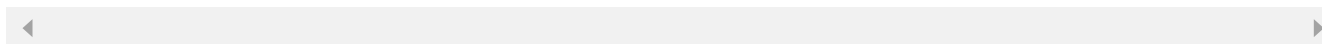
使用IDE执行程序

• Java堆溢出

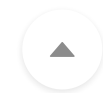
• VM arguments

设置Java堆的大小20MB，将堆的最小值-Xms参数和最大值-Xmx参数设置为一样，避免堆自动扩展。通过设置参数-XX:+HeapDumpOnOutOfMemoryError，在发生内存异常时可以生成堆转存储的dump文件，通过Eclipse Memory Analyzer (MAT)工具可以分析该文件。

```
1 VM Args:
2 -Xms20m -Xmx20m -XX:+HeapDumpOnOutOfMemoryError
```



Option	Description
-Xms20M -Xmx20M	to specify that a maximum heap size of 20 Mbytes
-XX:+HeapDumpOnOutOfMemoryError	Dump heap to file when java.lang.OutOfMemoryError is thrown



• Code

写下你的评论...



评论0



赞56



推荐阅读

最详细的JVM&GC讲解

阅读 10,215

JVM 内存模型 Stack Heap 文章选摘

阅读 1,824

Android内存管理分析总结

阅读 7,726

jQuery实现轮播

阅读 181

猎魔人

阅读 100

[JVM]OOM实例分析



伤口不该结疤

关注

赞赏支持

由 OutOfMemoryError 异常, 开始小及大堆内存的区域为 java heap space

```
1 import java.util.ArrayList;
2 import java.util.List;
3
4 /**
5  * VM Args:-Xms20m -Xmx20m -XX:+HeapDumpOnOutOfMemoryError
6  */
7 public class HeapOOM {
8     static class OOMObject {
9
10    }
11
12    public static void main(String[] args) {
13        List<OOMObject> list = new ArrayList<OOMObject>();
14
15        while (true) {
16            list.add(new OOMObject());
17        }
18    }
19 }
```

推荐阅读

最详细的JVM&GC讲解

阅读 10,215

JVM 内存模型 Stack Heap 文章选摘

阅读 1,824

Android内存管理分析总结

阅读 7,726

jQuery实现轮播

阅读 181

猎魔人

阅读 100

• Output

发生异常后生产的dump文件的默认名称为"java_pid进程id.hprof", 默认生成路径为工程的根目录

```
1 java.lang.OutOfMemoryError: Java heap space
2 Dumping heap to java_pid9512.hprof ...
```

写下你的评论...



评论0



赞56



[JVM]OOM实例分析



伤口不该结疤

关注

赞赏支持

```
8 |         at java.util.ArrayList.add(ArrayList.java:331)
9 |         at HeapOOM.main(HeapOOM.java:16)
```

使用MAT打开 java_pid9512.hprof, 可以生成一个分析报告

推荐阅读

最详细的JVM&GC讲解

阅读 10,215

JVM 内存模型 Stack Heap 文章选摘

阅读 1,824

Android内存管理分析总结

阅读 7,726

jQuery实现轮播

阅读 181

猎魔人

阅读 100

写下你的评论...



评论0



赞56



[JVM]OOM实例分析



伤口不该结疤

关注

赞赏支持

使用MAT打开 java_pid9512.hprof

点击进入Dominator Tree界面，可以将对象占用Heap大小进行排序，由下图可以看出main thread里面生成了很多HeapOOM\$OOMObject对象，导致暂用了97.49%的内存。

推荐阅读

最详细的JVM&GC讲解

阅读 10,215

JVM 内存模型 Stack Heap 文章选摘

阅读 1,824

Android内存管理分析总结

阅读 7,726

jQuery实现轮播

阅读 181

猎魔人

阅读 100



写下你的评论...



评论0



赞56



[JVM]OOM实例分析



伤口不该结疤

关注

赞赏支持

Dominator Tree

• 虚拟机栈溢出

• VM arguments

使用-Xss参数减少栈内存的容量

```
1 | VM Args:  
2 | -Xss128k
```

• Code

stackLeak()方法不断地自己调用自己，导致栈帧的深度一直在增加，当超出栈内存容量时，将会抛出java.lang.StackOverflowError异常

```
1 | /**  
2 |  * VM Args:-Xss128k  
3 |  */
```

推荐阅读

最详细的JVM&GC讲解

阅读 10,215

JVM 内存模型 Stack Heap 文章选摘

阅读 1,824

Android内存管理分析总结

阅读 7,726

jQuery实现轮播

阅读 181

猎魔人

阅读 100

写下你的评论...



评论0



赞56



[JVM]OOM实例分析



伤口不该结疤

关注

赞赏支持

```
10
11 /**
12  * @param args
13  * @throws Throwable
14  */
15 public static void main(String[] args) throws Throwable {
16     // TODO Auto-generated method stub
17     JavaVMStackSOF oom = new JavaVMStackSOF();
18     try {
19         oom.stackLeak();
20     } catch (Throwable e) {
21         // TODO: handle exception
22         System.out.println("stack length : " + oom.stackLength);
23         throw e;
24     }
25 }
26
27 }
28
```

推荐阅读

[最详细的JVM&GC讲解](#)

阅读 10,215

[JVM 内存模型 Stack Heap 文章选摘](#)

阅读 1,824

[Android内存管理分析总结](#)

阅读 7,726

[jQuery实现轮播](#)

阅读 181

[猎魔人](#)

阅读 100

• Output

```
1 stack length : 11432
2 Exception in thread "main" java.lang.StackOverflowError
3   at JavaVMStackSOF.stackLeak(JavaVMStackSOF.java:8)
4   at JavaVMStackSOF.stackLeak(JavaVMStackSOF.java:9)
5   at JavaVMStackSOF.stackLeak(JavaVMStackSOF.java:9)
6   ... ..
```

写下你的评论...



评论0



赞56



[JVM]OOM实例分析



伤口不该结疤

关注

赞赏支持

• VM arguments

通过-XX:PermSize和 -XX:MaxPermSize限制永久代的大小

```
1 | VM Args:  
2 | -XX:PermSize=10M -XX:MaxPermSize=10M
```

• Code

String.intern()的作用：如果字符串常量池中已经包含一个等于此String对象的字符串，则返回常量池中这个字符串的String对象；否则，将此String对象包含的字符串添加到常量池中，并且返回此String对象的引用。

下面这段代码一直在往常量池中添加字符串，直到超出常量池的容量。

```
1 | import java.util.ArrayList;  
2 | import java.util.List;  
3 |  
4 | /**  
5 |  * VM Args:-XX:PermSize=10M -XX:MaxPermSize=10M  
6 |  */  
7 | public class RuntimeConstantPoolOOM {  
8 |  
9 |     /**  
10 |      * @param args  
11 |      */  
12 |     public static void main(String[] args) {  
13 |         // TODO Auto-generated method stub
```

推荐阅读

最详细的JVM&GC讲解

阅读 10,215

JVM 内存模型 Stack Heap 文章选摘

阅读 1,824

Android内存管理分析总结

阅读 7,726

jQuery实现轮播

阅读 181

猎魔人

阅读 100

写下你的评论...



评论0



赞56



[JVM]OOM实例分析



伤口不该结疤

关注

赞赏支持

```
20  
21 }
```

推荐阅读

[最详细的JVM&GC讲解](#)

阅读 10,215

[JVM 内存模型 Stack Heap 文章选摘](#)

阅读 1,824

[Android内存管理分析总结](#)

阅读 7,726

[jQuery实现轮播](#)

阅读 181

[猎魔人](#)

阅读 100

• Output

- JDK 1.6及之前的版本中，常量池分配在永久代内，可以通过-XX:PermSize和 -XX:MaxPermSize限制永久代的大小，当常量池超出容量时，会抛出异常"Exception in thread "main" java.lang.OutOfMemoryError: PermGen space"
- 由于JDK 1.7的HotSpot中，已经把原本放在**永久代中的字符串常量池移除**，因此使用JDK 1.7运行这段代码时不会抛出异常，while循环将一直进行下去

写下你的评论...



评论0



赞56



[JVM]OOM实例分析



伤口不该结疤

关注

赞赏支持

JDK 1.6和JDK 1.7运行结果不同

• 本机直接内存溢出

• VM arguments

DirectMemory容量可以通过MaxDirectMemorySize指定，如果不指定，则默认与Java堆最大值一样(-Xmx指定)

```
1 VM Args:
2 -Xmx20M -XX:MaxDirectMemorySize=10M
```

• Code

通过Unsafe对象的allocateMemory方法申请直接内存，当超出DirectMemory容量时，抛出异常java.lang.OutOfMemoryError

```
1 import java.lang.reflect.Field;
2 import sun.misc.Unsafe;
3
4 /*
5  * VM Args: -Xmx20M -XX:MaxDirectMemorySize=10M
6  */
```

推荐阅读

最详细的JVM&GC讲解

阅读 10,215

JVM 内存模型 Stack Heap 文章选摘

阅读 1,824

Android内存管理分析总结

阅读 7,726

jQuery实现轮播

阅读 181

猎魔人

阅读 100

写下你的评论...



评论0



赞56



[JVM]OOM实例分析



伤口不该结疤

关注

赞赏支持

```
12     field unsafeField = Unsafe.class.getDeclaredField("unsafe");
13     unsafeField.setAccessible(true);
14     Unsafe unsafe = (Unsafe) unsafeField.get(null);
15     while (true) {
16         unsafe.allocateMemory(_1MB);
17     }
18 }
19 }
```

• Output

```
1 Exception in thread "main" java.lang.OutOfMemoryError
2     at sun.misc.Unsafe.allocateMemory(Native Method)
3     at DirectMemoryOOM.main(DirectMemoryOOM.java:16)
```

• 参考

- 深入理解Java虚拟机：JVM高级特性与最佳实践（第2版）
- [Configuring the Default JVM and Java Arguments](#)
- [Java HotSpot VM Options](#)



56人点赞 >



深入理解Java虚拟机学习笔记



推荐阅读

[最详细的JVM&GC讲解](#)

阅读 10,215

[JVM 内存模型 Stack Heap 文章选摘](#)

阅读 1,824

[Android内存管理分析总结](#)

阅读 7,726

[jQuery实现轮播](#)

阅读 181

[猎魔人](#)

阅读 100

写下你的评论...



评论0



赞56



[JVM]OOM实例分析



伤口不该结疤

关注

赞赏支持

还没有人赞赏，坚持一下



伤口不该结疤 Android开发

总资产319 (约28.78元) 共写了3.6W字 获得1,797个赞 共408个粉丝

关注

被以下专题收入，发现更多相似内容



JVM



Jvm使用



程序员



技术干货



深入JAVA虚拟机



深入理解Java虚拟机

推荐阅读

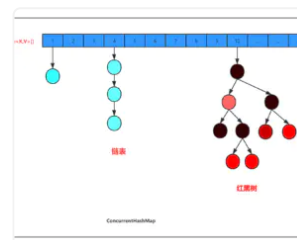
并发编程之ConcurrentHashMap源码解读-1.8

上一篇文章并发编程之synchronized的前世今生

[\[https://www.jianshu.com/p/849...\]](https://www.jianshu.com/p/849...)

默写流年 阅读 653 评论 2 赞 4

更多精彩内容>



写下你的评论...



评论0



赞56

...

[JVM]OOM实例分析



伤口不该结疤

关注

赞赏支持

短链接原理

说明 长链接比较长，一般分享有字数限制，所以需要转换为短链接。比如：长链接：[https://github.com/...](https://github.com/)

我是头像 愤怒的老照 阅读 42 评论 0 赞 0



java深度克隆工具类

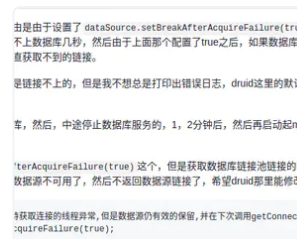
第一步：创建工具类 直接使用commons-beanutils实现对象拷贝，引入pom 创建工具类：BeanUtil...

flyjar 阅读 220 评论 0 赞 0

记一次因数据库重启导致Druid连接池GetConnectionTimeout...

问题描述：当线上有任务正在运行并且需要获取数据库连接的时候，此时重启了数据库(线上是hive)，导致线上服务后续一...

simples 阅读 226 评论 0 赞 0



推荐阅读

最详细的JVM&GC讲解

阅读 10,215

JVM 内存模型 Stack Heap 文章选摘

阅读 1,824

Android内存管理分析总结

阅读 7,726

jQuery实现轮播

阅读 181

猎魔人

阅读 100

写下你的评论...

评论0

赞56

...