

JVM基础及性能监控

(PDF测试岗位课程)

SUNING 苏宁



1

JVM基础知识

2

JVM垃圾回收机制

3

JVM内存和性能监控

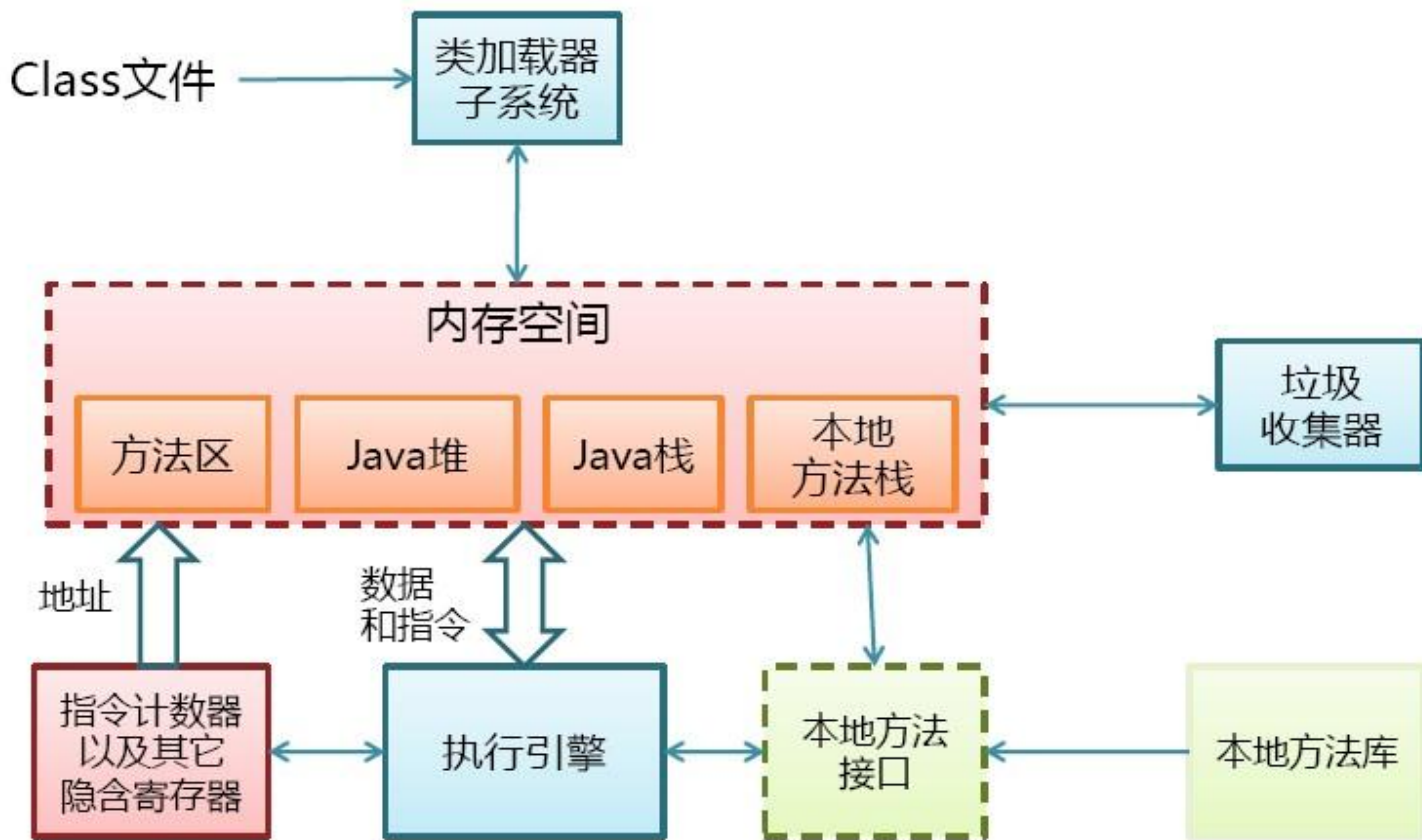
什么是JVM

JVM是Java Virtual Machine（Java虚拟机）的缩写，JVM是一种用于计算设备的规范，它是一个虚构出来的计算机，是通过在实际的计算机上仿真模拟各种计算机功能来实现的。

Java语言的一个非常重要的特点就是与平台的无关性。而使用Java虚拟机是实现这一特点的关键。一般的高级语言如果要在不同的平台上运行，至少需要编译成不同的目标代码。而引入Java语言虚拟机后，Java语言在不同平台上运行时不需要重新编译。Java语言使用Java虚拟机屏蔽了与具体平台相关的信息，使得Java语言编译程序只需生成在Java虚拟机上运行的目标代码（字节码），就可以在多种平台上不加修改地运行。Java虚拟机在执行字节码时，把字节码解释成具体平台上的机器指令执行。这就是Java的能够“一次编译，到处运行”的原因。

JVM内存组成结构

JVM内存由堆、栈、本地方法栈、方法区等部分组成，结构图如下所示：



启动

- 启动一个Java程序时，一个JVM实例就产生了，任何一个拥有 `public static void main(String[] args)` 函数的class都可以作为JVM实例运行的起点

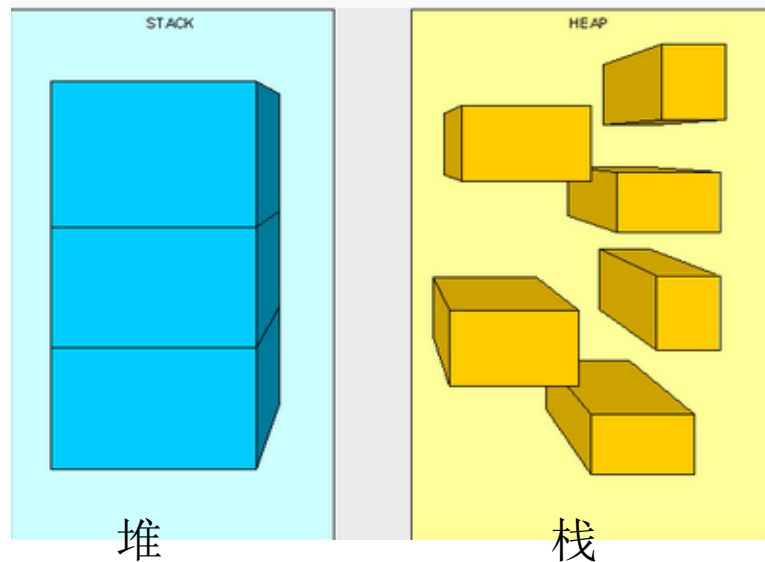
运行

- `main()`作为该程序初始线程的起点，任何其他线程均由该线程启动。JVM内部有两种线程：守护线程和非守护线程，`main()`属于非守护线程，守护线程通常由JVM自己使用，java程序也可以表明自己创建的线程是守护线程

消亡

- 当程序中的所有非守护线程都终止时，JVM才退出；若安全管理器允许，程序也可以使用`Runtime`类或者`System.exit()`来退出

栈 是运行时的单位，解决程序的运行问题，即程序如何执行，或者说如何处理数据；JVM栈是线程私有的，每个线程创建的同时都会创建JVM栈，JVM栈中存放的为当前线程中局部基本类型的变量（java中定义的八种基本类型：boolean、char、byte、short、int、long、float、double）、部分的返回结果以及Stack Frame，非基本类型的对象在JVM栈上仅存放一个指向堆上的地址。



JVM堆中存的是对象。JVM栈中存的是基本数据类型和JVM堆中对象的引用。一个对象的大小是不可估计的，或者说是可以动态变化的，但是在JVM栈中，一个对象只对应了一个4byte的引用

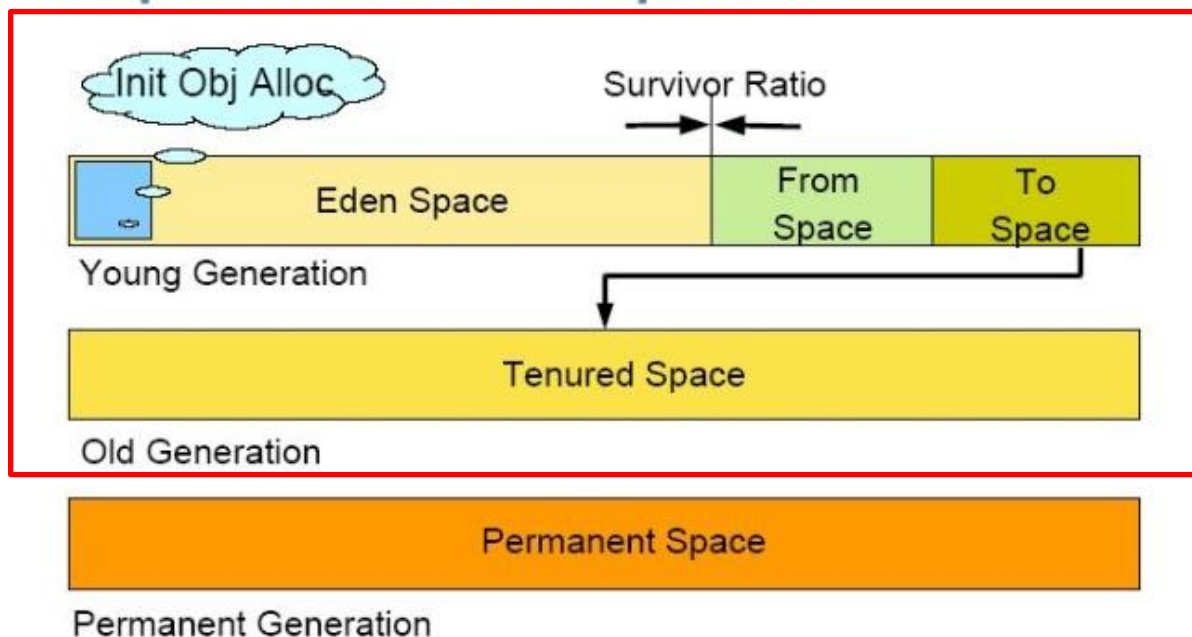
堆 是存储的单位，解决的是数据存储的问题，即数据怎么放、放在哪儿。它是JVM用来存储对象实例以及数组值的区域。堆是JVM中所有线程共享的。

JVM堆 (Heap) =

年轻代 (Young) + 年老代 (Tenured) + 持久代 (Permanent Generation)

年轻代 (Young) = 伊甸园区 (Eden) + 2个幸存者区 (Survivor)

Heap Structure of HotSpot VM



持久代主要存放的是Java类的类信息，JDK8版本以后把持久代移到了本地内存

年轻代:

所有新生成的对象首先都是放在年轻代的。年轻代的目标就是尽可能快速的收集掉那些生命周期短的对象。年轻代分三个区，即一个伊甸园（Eden）区，两个幸存者（Survivor）区（一般而言）。

年老代:

在年轻代中经历了N次垃圾回收后仍然存活的对象，就会被放到年老代中。因此，可以认为年老代中存放的都是一些生命周期较长的对象。

持久代:

用于存放静态文件，如今Java类、方法等。持久代对垃圾回收没有显著影响，但是有些应用可能动态生成或者调用一些class，在这种时候需要设置一个比较大的持久代空间来存放这些运行过程中新增的类。持久代大小通过-XX:MaxPermSize=<N>进行设置。

1

JVM基础知识

2

JVM垃圾回收机制

3

JVM内存和性能监控

为什么要垃圾回收

- 由于不同JAVA对象存活时间是不一定的，因此，在程序运行一段时间以后，如果不进行垃圾回收，整个程序会因内存耗尽导致整个程序崩溃。垃圾回收还会整理那些零散的内存碎片，碎片过多最直接的问题就是会导致无法分配大块的内存空间以及降低程序的运行效率

垃圾回收的概念和工作原理

- 概念：顾名思义，垃圾回收就是垃圾回收就是释放垃圾占用的空间，JVM中自动检测并移除不再使用的数据对象的这种机制称为垃圾回收，简称**GC**。
- 工作原理：JVM通过使用垃圾收集器及使用相应的垃圾回收算法将内存中不再被使用的对象进行回收。由于**GC**需要消耗一些资源和时间，**Java**在对对象的生命周期特征进行分析后，按照新生代、旧生代的方式来对对象进行收集，以尽可能的缩短**GC**对应用造成的暂停。

哪些内存可以垃圾回收

- JVM中会在以下情况触发回收：对象没有被引用，作用域发生未捕捉异常，程序正常执行完毕，程序执行了`System.exit()`，程序发生意外终止。



什么时候进行
垃圾回收？

当出现以下四种情况时，就会触发GC：

- 伊甸园区（Eden区）被写满
- 持久代（Perm）被写满
- System.gc()被显示调用
- 年老代（Tenured）被写满

基于GC作用范围，又把GC分为两种类型：

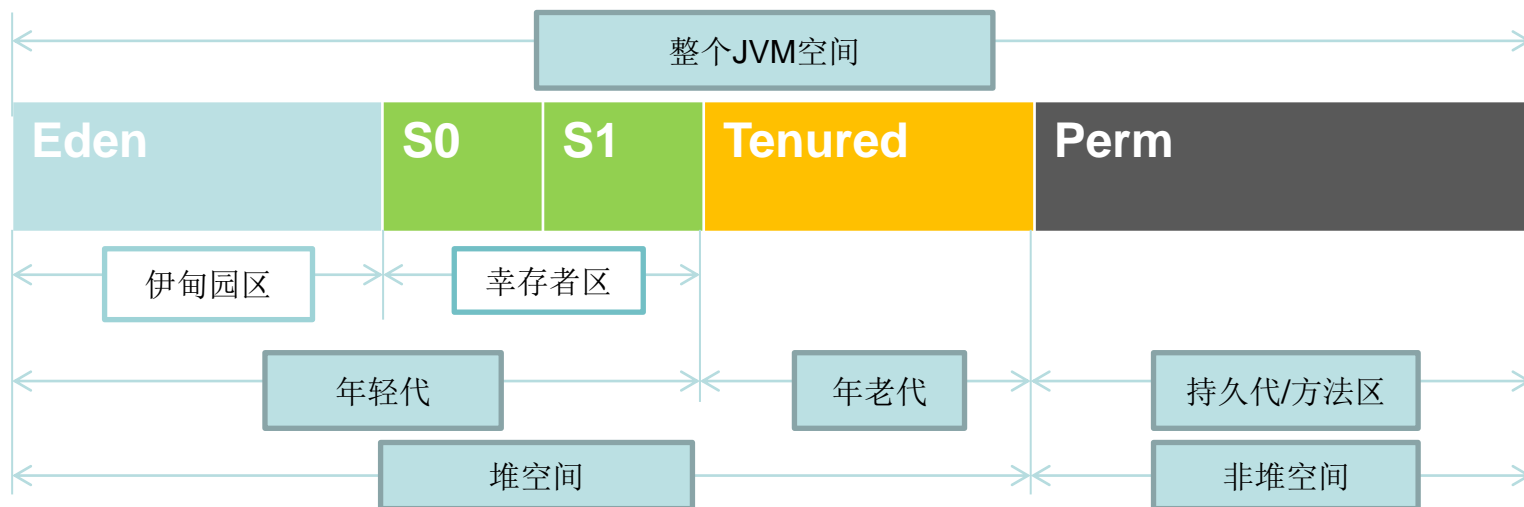
Scavenge GC和Full GC。

Scavenge GC负责年轻代整理，而Full GC则作用于整个堆。



JVM垃圾回收机制-Scavenge GC

SUNING 苏宁



Eden/S0/S1空间比例为Eden:S0:S1==8:1:1（可配置），有效内存（即可分配新生对象的内存）是总内存的9/10。

算法过程：

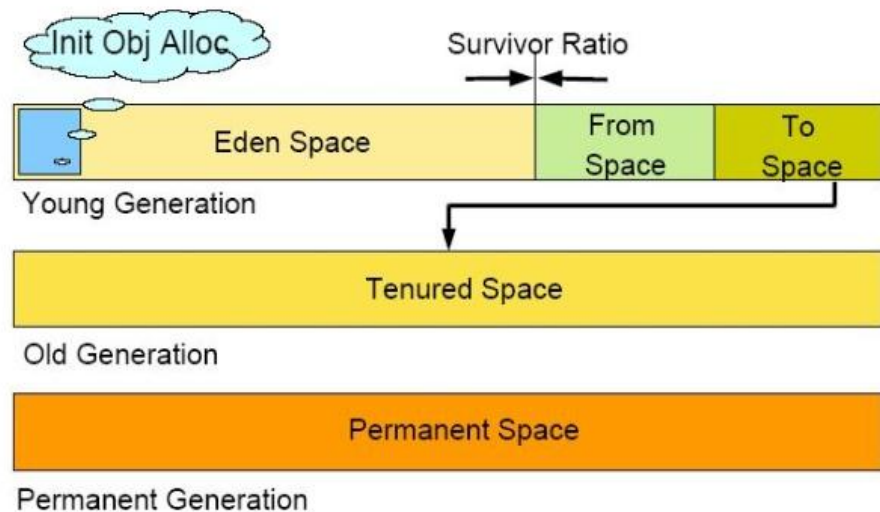
1. Eden可分配新生对象；
2. 对Eden+S0进行垃圾收集，存活对象复制到S1。清理Eden+S0。一次新生代GC结束。
3. Eden可分配新生对象；
4. 对Eden+S1进行垃圾收集，存活对象复制到S0。清理Eden+S1。二次新生代GC结束。
5. goto 1。

Full GC是对整个堆进行整理，包括新生代（Young）、旧生代（Tenured）、持久代（Permanent Generation）。Full GC比Scavenge GC要慢，因此应该尽可能减少Full GC。

有如下原因可能导致Full GC：

- o 年老代Tenured被写满
- o 持久代Perm域被写满
- o System.gc() 被显示调用
- o 上一次GC之后Heap的各域分配策略动态变化

Heap Structure of HotSpot VM



1

JVM基础知识

2

JVM垃圾回收机制

3

JVM内存和性能监控

JVM内存参数

```
java -Xmx3550m -Xms3550m -Xmn2g-Xss128k
```

-Xms

此参数控制 **Java** 堆的初始大小。调整此参数有助于降低垃圾回收开销，从而缩短服务器响应时间并提高吞吐量。对于某些应用程序，此选项的缺省设置可能会太低，这将导致发生大量小型垃圾回收。

-Xmx

此参数控制 **Java** 堆的最大大小。增大此参数将增加可供应用程序服务器使用的内存量，并且将降低垃圾回收频率。增大此设置可以缩短服务器响应时间并提高吞吐量。但是，增大此设置也将延长所执行的垃圾回收的持续时间。此设置不应大于可供应用程序服务器实例使用的系统内存量。如果将此设置增大到超出可用系统内存量，将导致发生系统页面调度，从而导致性能显著下降。

-Xmn

新生代的内存空间大小，**注意**：此处的大小是（eden+ 2 survivor space）。

-Xss:

设置每个线程的堆栈大小。JDK5.0以后每个线程堆栈大小为1M，以前每个线程堆栈大小为256K。更具应用的线程所需内存大小进行调整。在相同物理内存下，减小这个值能生成更多的线程。但是操作系统对一个进程内的线程数还是有限制的，不能无限生成，经验值在3000~5000左右

-XX:PermSize

设置非堆内存初始值，默认是物理内存的1/64

-XX:MaxPermSize

设置最大非堆内存的大小，默认是物理内存的1/4

-XX:NewSize=n:设置年轻代大小

-XX:NewRatio=n:设置年轻代和年老代的比值。如:为3，表示年轻代与年老代比值为1：3，年轻代占整个年轻代年老代和的1/4

-XX:SurvivorRatio=n:年轻代中Eden区与两个Survivor区的比值。注意Survivor区有两个。如：3，表示Eden：Survivor=3：2，一个Survivor区占整个年轻代的1/5

JDK监控工具: `jstack`, `jconsole`, `jinfo`, `jstat`, `JvisualVM`

Jstack

如果java程序崩溃生成core文件，jstack工具可以用来获得core文件的java stack和native stack的信息，从而可以轻松地知道java程序是如何崩溃和在程序何处发生问题。另外，jstack工具还可以附属到正在运行的java程序中，看到当时运行的java程序的java stack和native stack的信息，如果现在运行的java程序呈现hung的状态，jstack是非常有用的。

```
jstack pid > jstack.txt
```

Jconsole

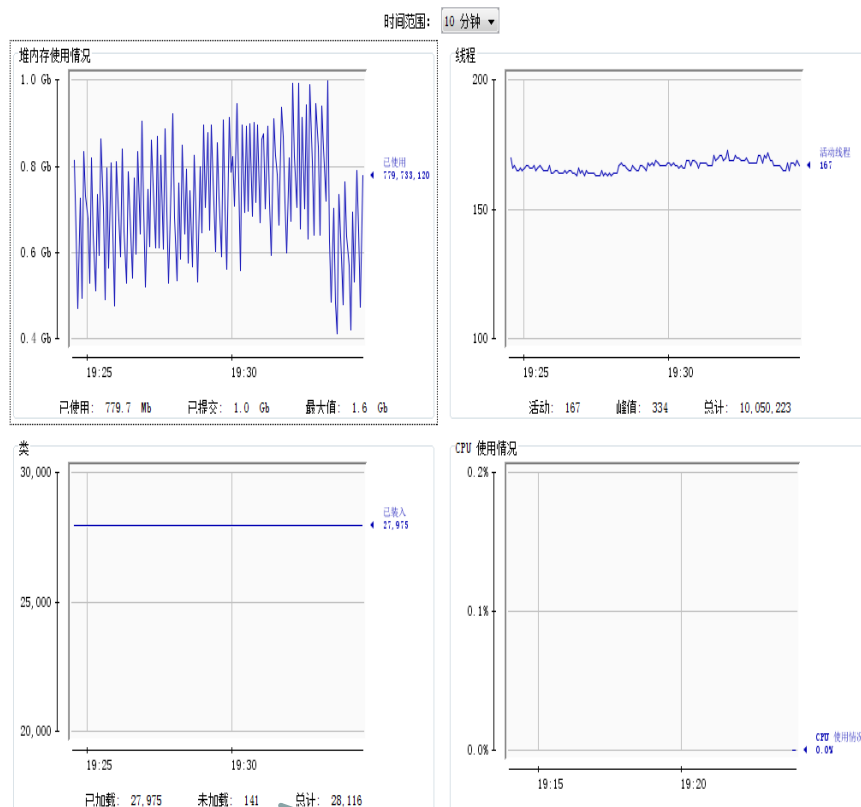
jconsole是基于Java Management Extensions (JMX)的实时图形化监测工具，这个工具利用了内建到JVM里面的JMX指令来提供实时的性能和资源的监控，包括了Java程序的内存使用，Heap size，线程的状态，类的分配状态和空间使用等等。

JDK监控工具

SUNING 苏宁

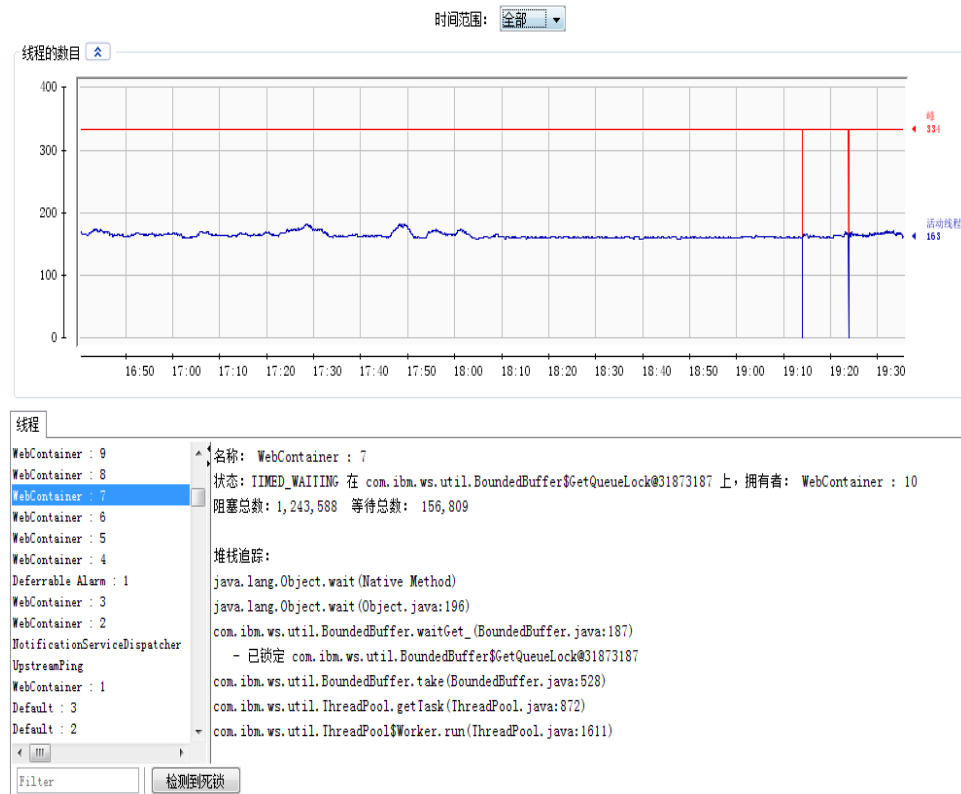


概述 内存 线程 类 VM 摘要 MBean



实时监控JVM内存
情况

概述 内存 线程 类 VM 摘要 MBean



进行线程的堆栈
追踪

jinfo

可以输出并修改运行时的java进程的参数

jinfo pid 会打印出详细的jvm运行参数和Java System属性。

jstat

jstat -gcutil: 统计gc时, heap情况

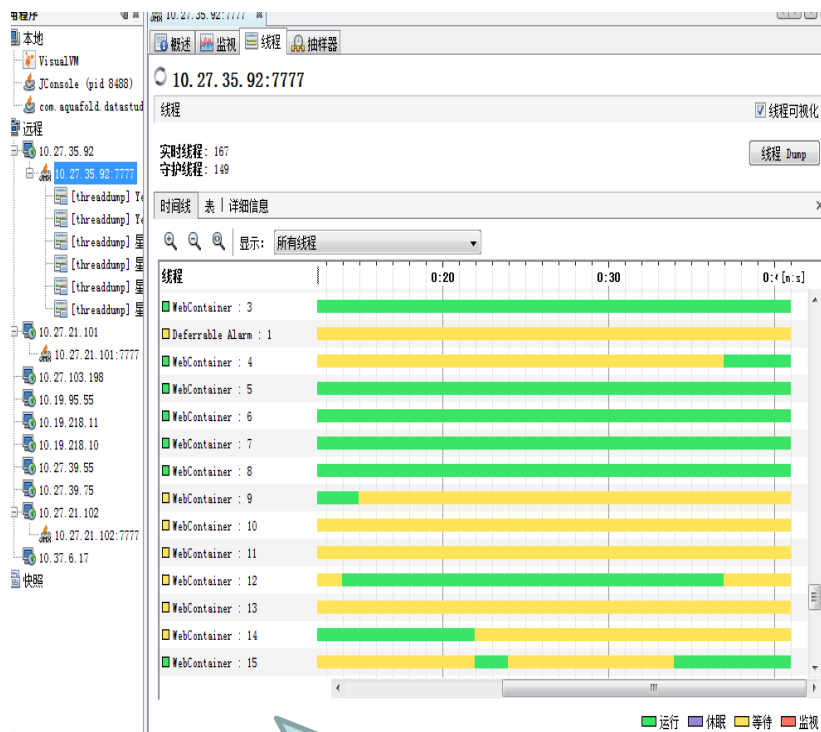
jstat -gc 垃圾回收统计

```
D:\jdk\bin>jstat -gcutil 5876 1000 18
```

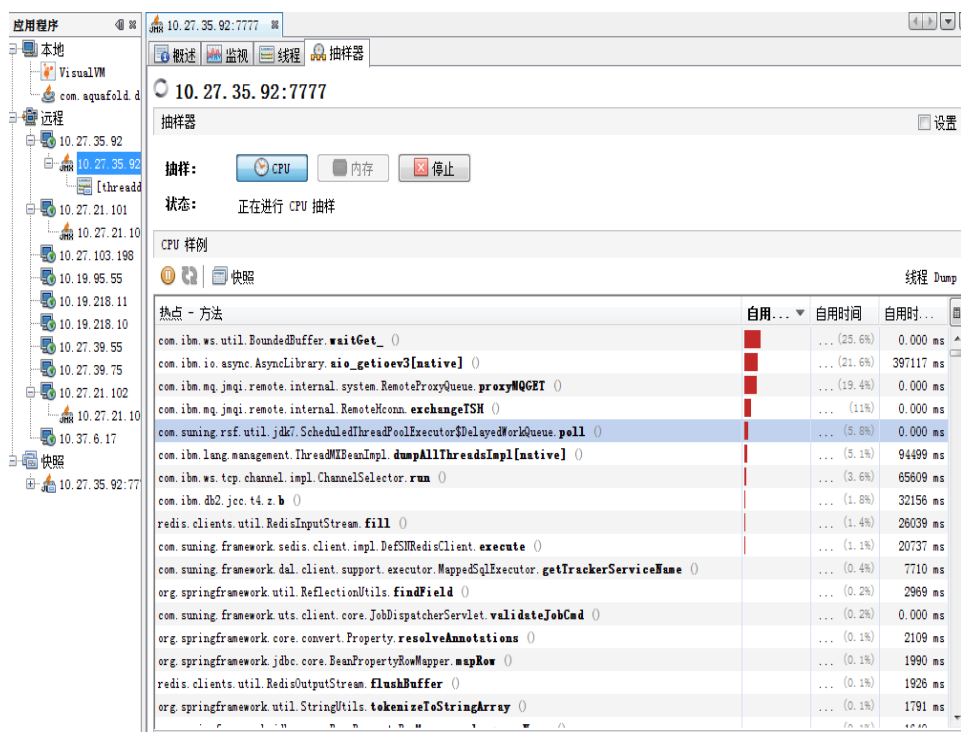
| S0 | S1 | E | O | P | YGC | YGCT | FGC | FGCT | GCT |
|-------|-------|-------|-------|-------|-----|-------|-----|-------|-------|
| 31.47 | 0.00 | 83.93 | 65.15 | 99.45 | 112 | 0.458 | 5 | 0.281 | 0.739 |
| 31.47 | 0.00 | 83.93 | 65.15 | 99.45 | 112 | 0.458 | 5 | 0.281 | 0.739 |
| 31.47 | 0.00 | 83.93 | 65.15 | 99.45 | 112 | 0.458 | 5 | 0.281 | 0.739 |
| 31.47 | 0.00 | 83.93 | 65.15 | 99.45 | 112 | 0.458 | 5 | 0.281 | 0.739 |
| 0.00 | 31.40 | 15.73 | 65.15 | 99.45 | 113 | 0.462 | 5 | 0.281 | 0.744 |
| 0.00 | 31.40 | 15.73 | 65.15 | 99.45 | 113 | 0.462 | 5 | 0.281 | 0.744 |
| 0.00 | 31.40 | 15.73 | 65.15 | 99.45 | 113 | 0.462 | 5 | 0.281 | 0.744 |
| 0.00 | 31.40 | 15.73 | 65.15 | 99.45 | 113 | 0.462 | 5 | 0.281 | 0.744 |
| 0.00 | 31.40 | 47.80 | 65.15 | 99.45 | 113 | 0.462 | 5 | 0.281 | 0.744 |

JvisualVM

一个以监控、显示本地或者远程服务器JVM工作情况, 进行性能调优的工具。借助VisualVM, 我们可以实现对JVM内存各个子池、CPU、垃圾收集器等方面进行监控, 从而发现程序代码中潜在的泄露点和配置问题



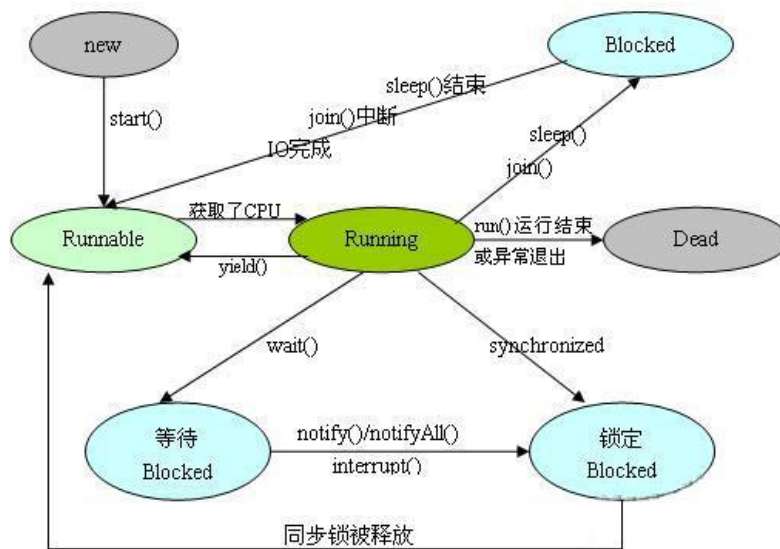
实时监控线程的运行状态，并且可以生成线程 Dump



查看热点方法CPU耗时，重点关注包含 suning的热点方法

线程状态

- 死锁（Deadlock）：一般指多个线程调用间，进入相互资源占用，导致一直等待无法释放的情况。
- 执行中（Runnable）：一般指该线程正在执行状态中，该线程占用了资源，正在处理某个请求，有可能在对某个文件操作，有可能进行数据类型等转换等。
- 等待资源（Waiting on condition）：等待资源，如果堆栈信息明确是应用代码，则证明该线程正在等待资源，一般是大量读取某资源、且该资源采用了资源锁的情况下，线程进入等待状态。又或者，正在等待其他线程的执行等。



- 阻塞 (Blocked)：指当前线程执行过程中，所需要的资源长时间等待却一直未能获取到，被容器的线程管理器标识为阻塞状态，可以理解为等待资源超时的线程。这种情况在应用的日志中，一般可以看到CPU 饥渴，或者某线程已执行了较长时间的信息。
- 停止 (Parked)：与字面意思不同，主要是指线程空闲时候的状态。如在线程池中，当线程被调用使用后再放入到池子中，则其状态变成了Parked
- 等待监控器检查资源 (Waiting on monitor)
- 暂停 (Suspended)
- 对象等待中 (Object.wait())

javacore文件分析思路

- Kill -3 pid 获取javacore文件，通过TMDA（IBM Thread and Monitor Dump Analyzer for Java）工具进行分析，重点关注阻塞（Blocked）、等待状态（Waiting on condition）和死锁（Deadlock）的线程

● Free Java heap size: 335.4 MB

● Allocated Java heap size: 1 GB

● Memory Segment Analysis

| Memory Type | # of Segments | Used Memory(bytes) | Used Memory(%) | Free Memory(bytes) | Free Memory(%) | Total Memory(bytes) |
|------------------|---------------|--------------------|----------------|--------------------|----------------|---------------------|
| Internal | 397 | 18,756,108 | 23.89 | 59,753,852 | 76.11 | 78,509,960 |
| Object(reserved) | 3 | 1,073,741,824 | 100 | 0 | 0 | 1,073,741,824 |
| Class | 3,773 | 129,500,388 | 82.86 | 26,796,032 | 17.14 | 156,296,420 |
| JIT Code Cache | 3 | 25,165,824 | 100 | 0 | 0 | 25,165,824 |
| JIT Data Cache | 2 | 9,152,944 | 54.56 | 7,624,272 | 45.44 | 16,777,216 |
| Overall | 4,178 | 1,256,317,088 | 93.03 | 94,174,156 | 6.97 | 1,350,491,244 |

Memory Segment Analysis

JVM内存
使用情况

JVM堆栈分析实例

SUNING 苏宁



| Thread Detail : javacore.20160428.113011.19029.0001.txt | | | | | |
|---|---------|----------|--------------|------------|--|
| Name | State | NativeID | Method | Stack D... | |
| I_Tch... | Wait... | 0x4ad9 | java/lan... | 3 | at java/util/Timer\$TimerImpl.(Timer.java:197(Compiled Code)) |
| I_Tch... | Wait... | 0x4af8 | java/lan... | 3 | at java/util/Timer.(Timer.java:350(Compiled Code)) |
| read... | Wait... | 0x4acd | java/lan... | 3 | at com/ibm/db2/jcc/am/gn.a(gn.java:4764(Compiled Code)) |
| =0:P... | Run... | 0x4ac1 | java/net/... | 6 | at com/ibm/db2/jcc/am/hn.b(hn.java:3580(Compiled Code)) |
| read... | Wait... | 0x4af3 | java/lan... | 3 | at com/ibm/db2/jcc/am/hn.ec(hn.java:720(Compiled Code)) |
| in-def... | Wait... | 0x4afd | java/lan... | 6 | at com/ibm/db2/jcc/am/hn.executeUpdate(hn.java:700(Compiled Code)) |
| calNo... | Wait... | 0x4b3f | java/lan... | 6 | at com/ibm/ws/rsadapter/jdbc/WJSJdbcPreparedStatement.pmiExecuteUpdate(WJSJdbcPreparedStatement.java:1(Compiled Code)) |
| in-def... | Wait... | 0x4afc | java/lan... | 6 | at com/ibm/ws/rsadapter/jdbc/WJSJdbcPreparedStatement.executeUpdate(WJSJdbcPreparedStatement.java:1(Compiled Code)) |
| MQJC... | Wait... | 0x4b1b | java/lan... | 4 | at org/springframework/jdbc/core/JdbcTemplate\$2.doInPreparedStatement(JdbcTemplate.java:818(Compiled Code)) |
| tach ... | Run... | 0x4a87 | com/lib... | 4 | at org/springframework/jdbc/core/JdbcTemplate\$2.doInPreparedStatement(JdbcTemplate.java:1(Compiled Code)) |
| ssag... | Wait... | 0x4ae8 | java/lan... | 3 | at org/springframework/jdbc/core/JdbcTemplate.execute(JdbcTemplate.java:587(Compiled Code)) |
| fault... | Wait... | 0x4b66 | java/lan... | 26 | at org/springframework/jdbc/core/JdbcTemplate.update(JdbcTemplate.java:812(Compiled Code)) |
| MQJC... | Wait... | 0x4b1a | java/lan... | 4 | at org/springframework/jdbc/core/JdbcTemplate.update(JdbcTemplate.java(Compiled Code)) |
| fault... | Wait... | 0x4b91 | java/lan... | 6 | at org/springframework/jdbc/core/namedparam/NamedParameterJdbcTemplate.update(NamedParameterJdbcTemplate.java(Compiled Code)) |
| fault... | Wait... | 0x4b8e | java/lan... | 6 | at org/springframework/jdbc/core/namedparam/NamedParameterJdbcTemplate.update(NamedParameterJdbcTemplate.java:264(Compiled Code)) |
| istrea... | Wait... | 0x4c35 | java/lan... | 3 | at com/suning/framework/dal/client/support/executor/MappedSqlExecutor.execute4PrimaryKey(MappedSqlExecutor.java:782(Compiled Code)) |
| PCCha... | Wait... | 0x4cb9 | java/lan... | 6 | at com/suning/framework/dal/client/support/executor/MappedSqlExecutor.execute(MappedSqlExecutor.java(Compiled Code)) |
| ebCo... | Wait... | 0x5577 | java/sec... | 104 | at com/suning/framework/dal/client/support/ShardingDalClient.execute(ShardingDalClient.java:281(Compiled Code)) |
| ebCo... | Wait... | 0x5279 | java/net/... | 71 | at com/suning/ebuy/spesv2/act/dao/impl/RewardSetDaoImpl.deleteFreeGift(RewardSetDaoImpl.java:177(Compiled Code)) |
| ebCo... | Wait... | 0x4efa | java/net/... | 95 | at com/suning/ebuy/spesv2/act/cart/service/impl/PersistRewardServiceImpl.persistReward(PersistRewardServiceImpl.java:108(Compiled Code)) |
| ared ... | Run... | 0x4af9 | sun/nio/... | 7 | at com/suning/ebuy/spesv2/act/cart/service/impl/PersistRewardServiceImpl\$FastClassByCGLIB\$\$8f2c25((Compiled Code)) |
| ebCo... | Wait... | 0x6175 | com/lib... | 107 | at net.sf.cglib/proxy/MethodProxy.invoke(MethodProxy.java:204(Compiled Code)) |
| rsion... | Wait... | 0x4b49 | java/lan... | 3 | |
| ra.net... | Run... | 0x4b8c | java/net/... | 5 | |
| ol ch... | Wait... | 0x4b4f | java/lan... | 3 | |
| ebCo... | Wait... | 0x238b | java/net/... | 98 | |
| fault... | Run... | 0x4b94 | com/lib... | 5 | |
| lMan... | Wait... | 0x4b2a | java/lan... | 6 | |
| ebCo... | Wait... | 0x4ef9 | java/net/... | 98 | |
| ebCo... | Wait... | 0x4f72 | java/lan... | 6 | |
| ol-2-t... | Park... | 0x4c7a | sun/mis... | 8 | |
| ferra... | Wait... | 0x4ab5 | java/lan... | 6 | |
| ebCo... | Wait... | 0x71c0 | java/nio/... | 62 | |
| fault... | Wait... | 0x4b62 | java/lan... | 23 | |
| Dae... | Wait... | 0x4b46 | java/lan... | 3 | |

等待db执行

JVM堆栈分析实例

SUNING 苏宁



| | | | |
|----------------|-------------------|-----------|-----|
| Thread-29 | Waiting on con... | 0...ja... | 4 |
| Thread-40 | Waiting on con... | 0...ja... | 4 |
| Thread-5 | Waiting on con... | 0...ja... | 3 |
| Thread-52 | Runnable | 0...ja... | 12 |
| Thread-53 | Runnable | 0...ja... | 12 |
| Thread-54 | Runnable | 0...ja... | 12 |
| Thread-7 | Waiting on con... | 0...ja... | 3 |
| ThreadMan... | Waiting on con... | 0...ja... | 3 |
| ThreadMan... | Waiting on con... | 0...ja... | 3 |
| ThreadMan... | Waiting on con... | 0...ja... | 3 |
| ThreadMan... | Waiting on con... | 0...ja... | 3 |
| JpstreamP... | Waiting on con... | 0...ja... | 3 |
| Version inf... | Waiting on con... | 0...ja... | 3 |
| WebContai... | Waiting on con... | 0...co... | 5 |
| WebCo... | Blocked | 0...ch... | 113 |
| WebContai... | Waiting on con... | 0...ja... | 124 |
| WebContai... | Waiting on con... | 0...ja... | 109 |
| WebContai... | Waiting on con... | 0...ja... | 6 |
| WebCo... | Waiting on con... | 0...ja... | 6 |
| WebCo... | Blocked | 0...ja... | 6 |
| WebCo... | Blocked | 0...ch... | 113 |
| WebContai... | Waiting on con... | 0...ja... | 6 |
| WebContai... | Waiting on con... | 0...ja... | 98 |
| WebCo... | Waiting on con... | 0...ch... | 113 |
| WebContai... | Waiting on con... | 0...ja... | 109 |
| WebCo... | Blocked | 0...ch... | 113 |
| WebContai... | Waiting on con... | 0...ja... | 6 |
| WebCo... | Blocked | 0...ch... | 113 |
| WebCo... | Blocked | 0...ch... | 136 |
| WebCo... | Blocked | 0...ja... | 6 |
| WebCo... | Blocked | 0...ch... | 113 |
| WebContai... | Runnable | 0...co... | 5 |
| WebContai... | Runnable | 0...co... | 5 |
| WebSpher... | Waiting on con... | 0...ja... | 3 |
| WLMMonit... | Waiting on con... | 0...ja... | 4 |

| Thread Name | Web Containe |
|---|--|
| State | Blocked |
| Monitor | Waiting for Monitor Lock on ch/qos/logback/core/spi/LogbackLock@0x000000007056250/0x00000000 |
| at ch/qos/logback/core/OutputStreamAppender.subAppend(OutputStreamAppender.java:211(Compiled Code)) | |
| at ch/qos/logback/core/rolling/RollingFileAppender.subAppend(RollingFileAppender.java:148(Compiled Code)) | |
| at com/suning/framework/log/appender/rolling/SnfRollingFileAppender.subAppend(SnfRollingFileAppender.java:103(Compiled Code)) | |
| at ch/qos/logback/core/OutputStreamAppender.append(OutputStreamAppender.java:103(Compiled Code)) | |
| at ch/qos/logback/core/UnsynchronizedAppenderBase.doAppend(UnsynchronizedAppenderBase.java:67(Compiled Code)) | |
| at ch/qos/logback/core/spi/AppenderAttachableImpl.appendLoopOnAppenders(AppenderAttachableImpl.java:113(Compiled Code)) | |
| at ch/qos/logback/classic/Logger.appendLoopOnAppenders(Logger.java:273(Compiled Code)) | |
| at ch/qos/logback/classic/Logger.callAppenders(Logger.java:273(Compiled Code)) | |
| at ch/qos/logback/classic/Logger.buildLoggingEventAndAppend(Logger.java:426(Compiled Code)) | |
| at ch/qos/logback/classic/Logger.filterAndLog_0_Or3Plus(Logger.java:396(Compiled Code)) | |
| at ch/qos/logback/classic/Logger.info(Logger.java:354(Compiled Code)) | |
| at com/suning/ebuy/spesv2/components/aspect/TimeCostAspect.logAround(TimeCostAspect.java:81(Compiled Code)) | |
| at sun.reflect.GeneratedMethodAccessor0.invoke(GeneratedMethodAccessor0.java:40(Compiled Code)) | |
| at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:37(Compiled Code)) | |
| at java.lang.reflect.Method.invoke(Method.java:589(Compiled Code)) | |
| at org.springframework.aop.aspectj.AbstractAspectJAdvice.invokeAdviceMethodWithGivenArgs(AbstractAspectJAdvice.java:645(Compiled Code)) | |
| at org.springframework.aop.aspectj.AbstractAspectJAdvice.invokeAdviceMethod(AbstractAspectJAdvice.java:625(Compiled Code)) | |
| at org.springframework.aop.aspectj.AspectJAroundAdvice.invoke(AspectJAroundAdvice.java:65(Compiled Code)) | |
| at org.springframework.aop.framework.ReflectiveMethodInvocation.proceed(ReflectiveMethodInvocation.java:173(Compiled Code)) | |
| at org.springframework.aop.interceptor.ExposeInvocationInterceptor.invoke(ExposeInvocationInterceptor.java:97(Compiled Code)) | |
| at org.springframework.aop.framework.ReflectiveMethodInvocation.proceed(ReflectiveMethodInvocation.java:173(Compiled Code)) | |
| at org.springframework.aop.framework.Cglib2AopProxy\$DynamicAdvisedInterceptor.intercept(Cglib2AopProxy\$DynamicAdvisedInterceptor.java:57(Compiled Code)) | |
| at com/suning/ebuy/spesv2/components/coupon/service/impl/CouponCheckAvailableServiceImpl\$\$Enhancer\$1.doCheck(CouponCheckAvailableServiceImpl\$\$Enhancer\$1.java:11) | |
| at com/suning/ebuy/spesv2/components/actsearch/service/impl/SearchCommdtyCouponActServiceImpl.ch... | |

打印耗时日志阻塞

"WebContainer : 5" - Thread t@118

java.lang.Thread.State: **BLOCKED**

at java.util.Hashtable.get(Hashtable.java:518)

- waiting to lock <2590259> (a java.util.Hashtable) owned by "WebContainer : 0" t@113

at java.lang.Class.lookupCachedFields(Class.java:2543)

at java.lang.Class.getDeclaredFields(Class.java:546)

at org.codehaus.jackson.map.introspect.AnnotatedClass._addFields(AnnotatedClass.java:697)

at org.codehaus.jackson.map.introspect.AnnotatedClass.resolveFields(AnnotatedClass.java:457)

at org.codehaus.jackson.map.introspect.BasicClassIntrospector.collectProperties(BasicClassIntrospector.java:100)

at org.codehaus.jackson.map.introspect.BasicClassIntrospector.forSerialization(BasicClassIntrospector.java:130)

at org.codehaus.jackson.map.introspect.BasicClassIntrospector.forSerialization(BasicClassIntrospector.java:130)

at org.codehaus.jackson.map.SerializationConfig.introspect(SerializationConfig.java:973)

at org.codehaus.jackson.map.ser.BeanSerializerFactory.createSerializer(BeanSerializerFactory.java:110)

at org.codehaus.jackson.map.ser.StdSerializerProvider._createUntypedSerializer(StdSerializerProvider.java:100)

at org.codehaus.jackson.map.ser.StdSerializerProvider._createAndCacheUntypedSerializer(StdSerializerProvider.java:100)

at org.codehaus.jackson.map.ser.StdSerializerProvider.findValueSerializer(StdSerializerProvider.java:100)

at org.codehaus.jackson.map.ser.StdSerializerProvider.findTypedValueSerializer(StdSerializerProvider.java:100)

at org.codehaus.jackson.map.ser.StdSerializerProvider._serializeValue(StdSerializerProvider.java:100)

at org.codehaus.jackson.map.ser.StdSerializerProvider.serializeValue(StdSerializerProvider.java:100)

at org.codehaus.jackson.map.ObjectMapper.writeValue(ObjectMapper.java:1613)

Json使用
不当

- JVM内存结构
- 栈是运行时的单位，而堆是存储的单位
- JVM堆分为年轻代、年老代和持久代（JDK8版本以后从堆内存移到本地内存），年轻代分为伊甸园区（Eden）和2个幸存者区（Survivor）
- JVM垃圾回收机制（Scavenge GC和Full GC）
- JDK监控工具
- 线程状态和JAVACORE分析

THANK YOU

谢 谢

SUNING 苏宁

