(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 110888712 A
(43)申请公布日 2020.03.17

(21)申请号 201910960132.1

(22)申请日 2019.10.10

(71)申请人 望海康信(北京)科技股份公司
地址 100176 北京市大兴区北京经济技术开发区荣华中路22号院3号楼8层801-2

(72)发明人 胡阳辉

(74)专利代理机构 北京金阙华进专利事务所(普通合伙) 11224
代理人 陈建春

(51)Int.Cl.
G06F 9/455(2006.01)
G06F 9/50(2006.01)

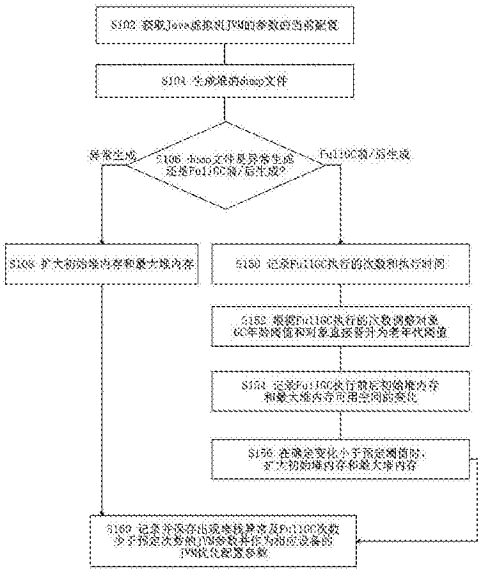
权利要求书2页 说明书5页 附图2页

(54)发明名称

Java虚拟机优化方法及系统

(57)摘要

本申请公开了Java虚拟机优化方法及系统，其中所述方法包括：获取Java虚拟机JVM的参数的当前配置，所述参数包括初始堆内存、最大堆内存、新生代与老年代的比例、对象GC年龄阈值和对象直接晋升为老年代阈值；确定所生成的大小堆dump文件是异常生成的dump文件还是FullGC前/后生成的dump文件；响应于确定dump文件是异常生成的dump文件，重新配置初始堆内存、最大堆内存和对象GC年龄阈值；响应于确定dump文件是FullGC前/后生成的dump文件，调整新生代与老年代的比例、对象GC年龄阈值和对象直接晋升为老年代阈值。本发明使能节省开发时间，提高开发效率。



CN 110888712 A

1.一种Java虚拟机优化方法,其特征在于,所述方法包括:

获取Java虚拟机JVM的参数的当前配置,所述参数包括初始堆内存、最大堆内存、新生代与老年代的比例、对象GC年龄阈值和对象直接晋升为老年代阈值;

确定所生成的大小堆dump文件是异常生成的dump文件还是FullGC前/后生成的dump文件;

响应于确定dump文件是异常生成的dump文件,重新配置初始堆内存、最大堆内存和对象GC年龄阈值;

响应于确定dump文件是FullGC前/后生成的dump文件,调整新生代与老年代的比例、对象GC年龄阈值和对象直接晋升为老年代阈值。

2.根据权利要求1所述的方法,其特征在于,所述响应于确定dump文件是FullGC前、后生成的dump文件,调整新生代与老年代的比例、对象GC年龄阈值和对象直接晋升为老年代阈值包括:

记录FullGC执行的次数和执行时间;

根据FullGC执行的次数调整对象GC年龄阈值和对象直接晋升为老年代阈值,FullGC执行的次数越多,对象GC年龄阈值和对象直接晋升为老年代阈值越小。

3.根据权利要求2所述的方法,其特征在于,所述方法还包括:

记录FullGC执行前后初始堆内存和最大堆内存可用空间的变化;

响应于所述变化小于预定阈值,扩大初始堆内存和最大堆内存。

4.根据权利要求1所述的方法,所述方法还包括:

记录并保存出现堆栈异常次数少于第一预定次数及FullGC次数少于第二预定次数的JVM参数并作为相应设备的JVM优化配置参数。

5.根据权利要求1所述的方法,所述方法还包括:

响应于空余堆内存小于第一阈值,增大初始堆内存;及

响应于空余堆内存大于第二阈值,减小最大堆内存。

6.根据权利要求4所述的方法,所述方法还包括:

响应于确定新设备与JVM优化配置参数所对应的设备为同样或同类设备,将所述JVM优化配置参数作为新设备的初始JVM配置参数。

7.根据权利要求1所述的方法,其中所述响应于确定dump文件是异常生成的dump文件,重新配置初始堆内存、最大堆内存和对象GC年龄阈值包括:

增大初始堆内存和最大堆内存;及

减小对象GC年龄阈值。

8.一种Java虚拟机优化系统,其特征在于,所述系统包括:

当前配置获取模块,用于获取Java虚拟机JVM的参数的当前配置,所述参数包括初始堆内存、最大堆内存、新生代与老年代的比例、对象GC年龄阈值和对象直接晋升为老年代阈值;

文件类型确定模块,用于确定所生成的大小堆dump文件是异常生成的dump文件还是FullGC前/后生成的dump文件;

重新配置模块,用于响应于确定dump文件是异常生成的dump文件,重新配置初始堆内存、最大堆内存和对象GC年龄阈值;

调整模块，用于响应于确定dump文件是FullGC前/后生成的dump文件，调整新生代与老年代的比例、对象GC年龄阈值和对象直接晋升为老年代阈值。

9.根据权利要求8所述的系统，所述系统还包括：

记录模块，用于记录并保存出现堆栈异常次数少于第一预定次数及FullGC次数少于第二预定次数的JVM参数并作为相应设备的JVM优化配置参数。

10.根据权利要求9所述的系统，所述系统还包括：

初始化模块，用于响应于确定新设备与JVM优化配置参数所对应的设备为同样或同类设备，将所述JVM优化配置参数作为新设备的初始JVM配置参数。

Java虚拟机优化方法及系统

技术领域

[0001] 本申请涉及电数字数据处理领域，尤其涉及Java虚拟机优化方法及系统。

背景技术

[0002] UNIEAP是东软集团开发的开发平台。对于UNIEAP初学开发者而言，内存溢出是十分常见的现象。对于不同设备，Java虚拟机(JVM)的最佳参数配置亦各有不同。

[0003] 目前，对于UNIEAP的初学开发者，由于UNIEAP版本的不同以及设备的参数不同，造成JVM大小堆不尽相同。开发过程中，新生代、老年代、持久代的对象的创建和回收策略各不相同，GC回收(垃圾回收)不及时或不能完全回收，容易造成内存的堆栈溢出。

[0004] 因而需要一种避免UNIEAP初学者在JVM参数配置时花费过多时间的解决方案。

发明内容

[0005] 为了克服现有技术中存在的不足，本发明要解决的技术问题是提供一种Java虚拟机优化方法及系统，其使能节省开发者的时间，提高开发效率。

[0006] 为解决上述技术问题，根据本发明的第一方面，提供一种Java虚拟机优化方法，该方法包括：

[0007] 获取Java虚拟机JVM的参数的当前配置，所述参数包括初始堆内存、最大堆内存、新生代与老年代的比例、对象GC年龄阈值和对象直接晋升为老年代阈值；

[0008] 确定所生成的大小堆dump文件是异常生成的dump文件还是FullGC前/后生成的dump文件；

[0009] 响应于确定dump文件是异常生成的dump文件，重新配置初始堆内存、最大堆内存和对象GC年龄阈值；

[0010] 响应于确定dump文件是FullGC前/后生成的dump文件，调整新生代与老年代的比例、对象GC年龄阈值和对象直接晋升为老年代阈值。

[0011] 作为本发明所述方法的改进，所述响应于确定dump文件是FullGC前、后生成的dump文件，调整新生代与老年代的比例、对象GC年龄阈值和对象直接晋升为老年代阈值包括：记录FullGC执行的次数和执行时间；根据FullGC执行的次数调整对象GC年龄阈值和对象直接晋升为老年代阈值，FullGC执行的次数越多，对象GC年龄阈值和对象直接晋升为老年代阈值越小。

[0012] 作为本发明所述方法的另一种改进，所述方法还包括：记录FullGC执行前后初始堆内存和最大堆内存可用空间的变化；响应于所述变化小于预定阈值，扩大初始堆内存和最大堆内存。

[0013] 作为本发明所述方法的又一种改进，所述方法还包括：记录并保存出现堆栈异常次数少于第一预定次数及FullGC次数少于第二预定次数的JVM参数并作为相应设备的JVM优化配置参数。

[0014] 作为本发明所述方法的再一种改进，所述方法还包括：响应于空余堆内存小于第

一阈值，增大初始堆内存；及响应于空余堆内存大于第二阈值，减小最大堆内存。

[0015] 作为本发明所述方法的另一种改进，所述方法还包括：响应于确定新设备与JVM优化配置参数所对应的设备为同样或同类设备，将所述JVM优化配置参数作为新设备的初始JVM配置参数。

[0016] 作为本发明所述方法的又一种改进，其中所述响应于确定dump文件是异常生成的dump文件，重新配置初始堆内存、最大堆内存和对象GC年龄阈值包括：增大初始堆内存和最大堆内存；及减小对象GC年龄阈值。

[0017] 为解决上述技术问题，根据本发明的第二方面，提供一种Java虚拟机优化系统，该系统包括：

[0018] 当前配置获取模块，用于获取Java虚拟机JVM的参数的当前配置，所述参数包括初始堆内存、最大堆内存、新生代与老年代的比例、对象GC年龄阈值和对象直接晋升为老年代阈值；

[0019] 文件类型确定模块，用于确定所生成的大小堆dump文件是异常生成的dump文件还是FullGC前/后生成的dump文件；

[0020] 重新配置模块，用于响应于确定dump文件是异常生成的dump文件，重新配置初始堆内存、最大堆内存和对象GC年龄阈值；

[0021] 调整模块，用于响应于确定dump文件是FullGC前/后生成的dump文件，调整新生代与老年代的比例、对象GC年龄阈值和对象直接晋升为老年代阈值。

[0022] 为解决上述技术问题，本发明的有形计算机可读介质，包括用于执行本发明的Java虚拟机优化方法的计算机程序代码。

[0023] 为解决上述技术问题，本发明提供一种装置，包括至少一个处理器；及至少一个存储器，含有计算机程序代码，所述至少一个存储器和所述计算机程序代码被配置为利用所述至少一个处理器使得所述装置执行本发明的Java虚拟机优化方法的至少部分步骤。

[0024] 按照本发明，可根据dump文件情况实时调整jvm参数，使得开发过程更加便利，节约时间，从而提高尤其是初学开发者的开发效率。另外，通过记录不同设备的最佳参数配置，可以根据过往的设备参数初始化新设备参数配置，同样节省开发时间。

[0025] 结合附图阅读本发明实施方式的详细描述后，本发明的其它特点和优点将变得更加清楚。

附图说明

[0026] 图1为根据本发明方法的一实施例的流程图。

[0027] 图2为根据本发明系统的一实施例的结构示意图。

[0028] 为清晰起见，这些附图均为示意性及简化的图，它们只给出了对于理解本发明所必要的细节，而省略其他细节。

具体实施方式

[0029] 下面参照附图对本发明的实施方式和实施例进行详细说明。

[0030] 通过下面给出的详细描述，本发明的适用范围将显而易见。然而，应当理解，在详细描述和具体例子表明本发明优选实施例的同时，它们仅为说明目的给出。

[0031] 图1示出了根据本发明的Java虚拟机优化方法的一优选实施例的流程图。

[0032] 在步骤S102，获取Java虚拟机JVM的参数的当前配置，JVM参数包括但不限于初始堆内存-Xms、最大堆内存-Xmx、新生代与老年代的比例-XX:NewRatio、对象GC年龄阈值-XX:MaxTenuringThreshold和对象直接晋升为老年代阈值-XX:PretenureSizeThreshold。

[0033] JVM具体参数配置可被记录。

[0034] 软件运行时，通过脚本(Windows,Mac,Linux可能需要不同的脚本，脚本的执行可嵌入软件也可以手动执行)获取设备的配置参数(如:Windows:32位/64位、物理内存、CPU信息)。

[0035] JVM参数可根据设备的配置参数进行初始化配置或者结合大数据设定与当前设备相当的设备的最优JVM参数配置作为初始配置。

[0036] JVM通用参数配置如下：

[0037] 大小堆设置

[0038] -Xms:初始堆内存。默认物理内存的1/64，也是最小分配堆内存。当空余堆内存小于40％时，会增加到-Xms的最大限制；

[0039] -Xmx:最大堆内存。默认物理内存的1/4，当空余堆内存大于70％时，会减小到-Xms的最小限制。

[0040] 一般情况下，开发者在初始化时会把-Xmx和-Xms大小设置为一样，这样虚拟机不会再自动增大或减小内存占用，虚拟机相对稳定。如果这两个参数不一样，那么虚拟机会先获取-Xms的最小内存，当内存不够用时，会逐步申请扩大内存，最大为-Xmx，当虚拟机卸载一部分程序(即GC回收起了作用)，那么一部分使用的内存会变为可用，当这个空余内存大于70％时，为了节省设备内存，会自动优化，减少内存占用，重新降至-Xms。

[0041] -XX:PermSize:非堆内存的初始值，默认物理内存的1/64，也是最小非堆内存；

[0042] -XX:MaxPermSize:非堆内存最大值，默认物理内存的1/4。

[0043] 新生代设置

[0044] -Xmn:新生代大小(默认设置:Eden:Survivor＝8:1。一经设置，一般不再修改)。通常为Xmx的1/3或1/4。新生代＝Eden+2个Survivor空间。实际可用空间为＝Eden+1个Survivor，即90％。

[0045] 老年代设置

[0046] -XX:NewRatio:新生代与老年代的比例，如-XX:NewRatio＝2，则新生代占整个堆空间的1/3，老年代占2/3。

[0047] GC控制设置

[0048] -XX:MaxTenuringThreshold:对象GC年龄阈值；

[0049] -XX:PretenureSizeThreshold:对象直接晋升为老年代阈值(大于该值会直接晋升为老年代，避免频繁的minor GC)。

[0050] Dump文件设置

[0051] -XX:+HeapDumpOnOutOfMemoryError:JVM在异常时会自动生成dump文件；

[0052] -XX:+HeapDumpBeforeFullGC:实现在Full GC前dump；

[0053] -XX:+HeapDumpAfterFullGC:实现在Full GC后dump；

[0054] -XX:HeapDumpPath＝/exception/dump/date/:dump文件存储路径。

[0055] 在步骤S104，生成堆的dump文件。通过上面的参数配置控制dump文件的生成和存储位置，不同的dump文件存储在不同的位置。

[0056] 在步骤S106，根据dump文件的存储位置确定所生成的大小堆dump文件是异常生成的dump文件还是FullGC前/后生成的dump文件。如果是异常生成的dump文件，处理进行到步骤S108；否则，如果是FullGC前/后生成的dump文件，处理进行到步骤S150。

[0057] 在步骤S108，扩大初始堆内存-Xms和最大堆内存-Xmx的值，例如每次扩大256或者512兆，同时减小对象GC年龄阈值(分代年龄)，例如每次减小2。

[0058] 在步骤S150，记录FullGC执行的次数和执行时间。

[0059] 在步骤S152，根据FullGC执行的次数调整对象GC年龄阈值和对象直接晋升为老年代阈值，FullGC执行的次数越多，对象GC年龄阈值和对象直接晋升为老年代阈值越小。FullGC次数通常以每日不超过2次或者1次为佳，但并不限于此，根据项目大小、设备配置，FullGC次数越小越好。如果FullGC后，释放出了较多的内存空间，说明老年代对象较多，这时适当增大GC年龄，增大老年代内存大小(NewRatio)，增大晋升为老年代阈值(PretenureSizeThreshold)。

[0060] 在步骤S154，记录FullGC执行前后初始堆内存和最大堆内存可用空间的变化。FullGC后，如果释放较多内存空间，则增大-Xms和-Xmx这两个参数，同时增大GC分代年龄和直接晋升为老年代对象的阈值；反之则说明目前的内存是正常的。

[0061] 在步骤S156，在确定FullGC执行前后初始堆内存和最大堆内存可用空间的变化小于预定阈值时，扩大初始堆内存和最大堆内存。

[0062] 在步骤S160，记录并保存出现堆栈异常次数少于第一预定次数及FullGC次数少于第二预定次数的JVM参数并作为相应设备的JVM优化配置参数。例如，在某一次配置后，如果至少一周或一个月不出现异常，一天或一周FullGC次数很少比如不出现或者1次，可认为当前JVM参数是当前设备的优化配置参数。

[0063] 在其它实施例中，本发明方法还可包括，比对新设备的设备参数与已记录设备的设备参数的比较，根据新设备的设备参数，找到相同或类似设备参数的设备的JVM优化配置参数并将其作为新设备的初始JVM配置参数，以节约开发时间。

[0064] 图2示出了根据本发明的Java虚拟机优化系统的一实施例的框图，该系统包括：当前配置获取模块202，用于获取Java虚拟机JVM的参数的当前配置，所述参数包括初始堆内存、最大堆内存、新生代与老年代的比例、对象GC年龄阈值和对象直接晋升为老年代阈值；文件类型确定模块204，用于确定所生成的大小堆dump文件是异常生成的dump文件还是FullGC前/后生成的dump文件；重新配置模块206，用于响应于确定dump文件是异常生成的dump文件，重新配置初始堆内存、最大堆内存和对象GC年龄阈值；调整模块208，用于响应于确定dump文件是FullGC前/后生成的dump文件，调整新生代与老年代的比例、对象GC年龄阈值和对象直接晋升为老年代阈值。

[0065] 在其它实施例中，本发明系统还可包括：记录模块，用于记录并保存出现堆栈异常次数少于第一预定次数及FullGC次数少于第二预定次数的JVM参数并作为相应设备的JVM优化配置参数；和/或初始化模块，用于响应于确定新设备与JVM优化配置参数所对应的设备为同样或同类设备，将所述JVM优化配置参数作为新设备的初始JVM配置参数。

[0066] 在此所述的多个不同实施例或者其特定特征、结构或特性可在本发明的一个或多

个实施方式中适当组合。另外，在某些情形下，只要适当，流程图中和/或流水处理描述的步骤顺序可修改，并不必须精确按照所描述的顺序执行。另外，本发明的多个不同方面可使用软件、硬件、固件或者其组合和/或执行所述功能的其它计算机实施的模块或装置进行实施。本发明的软件实施可包括保存在计算机可读介质中并由一个或多个处理器执行的可执行代码。计算机可读介质可包括计算机硬盘驱动器、ROM、RAM、闪存、便携计算机存储介质如CD-ROM、DVD-ROM、闪盘驱动器和/或例如具有通用串行总线(USB)接口的其它装置，和/或任何其它适当的有形或非短暂计算机可读介质或可执行代码可保存于其上并由处理器执行的计算机存储器。本发明可结合任何适当的操作系统使用。

[0067] 除非明确指出，在此所用的单数形式“一”、“该”均包括复数含义(即具有“至少一”的意思)。应当进一步理解，说明书中使用的术语“具有”、“包括”和/或“包含”表明存在所述的特征、步骤、操作、元件和/或部件，但不排除存在或增加一个或多个其他特征、步骤、操作、元件、部件和/或其组合。如在此所用的术语“和/或”包括一个或多个列举的相关项目的任何及所有组合。

[0068] 前面说明了本发明的一些优选实施例，但是应当强调的是，本发明不局限于这些实施例，而是可以本发明主题范围内的其它方式实现。本领域技术人员可以在本发明技术构思的启发和不脱离本发明内容的基础上对本发明做出各种变型和修改，这些变型或修改仍落入本发明的保护范围之内。

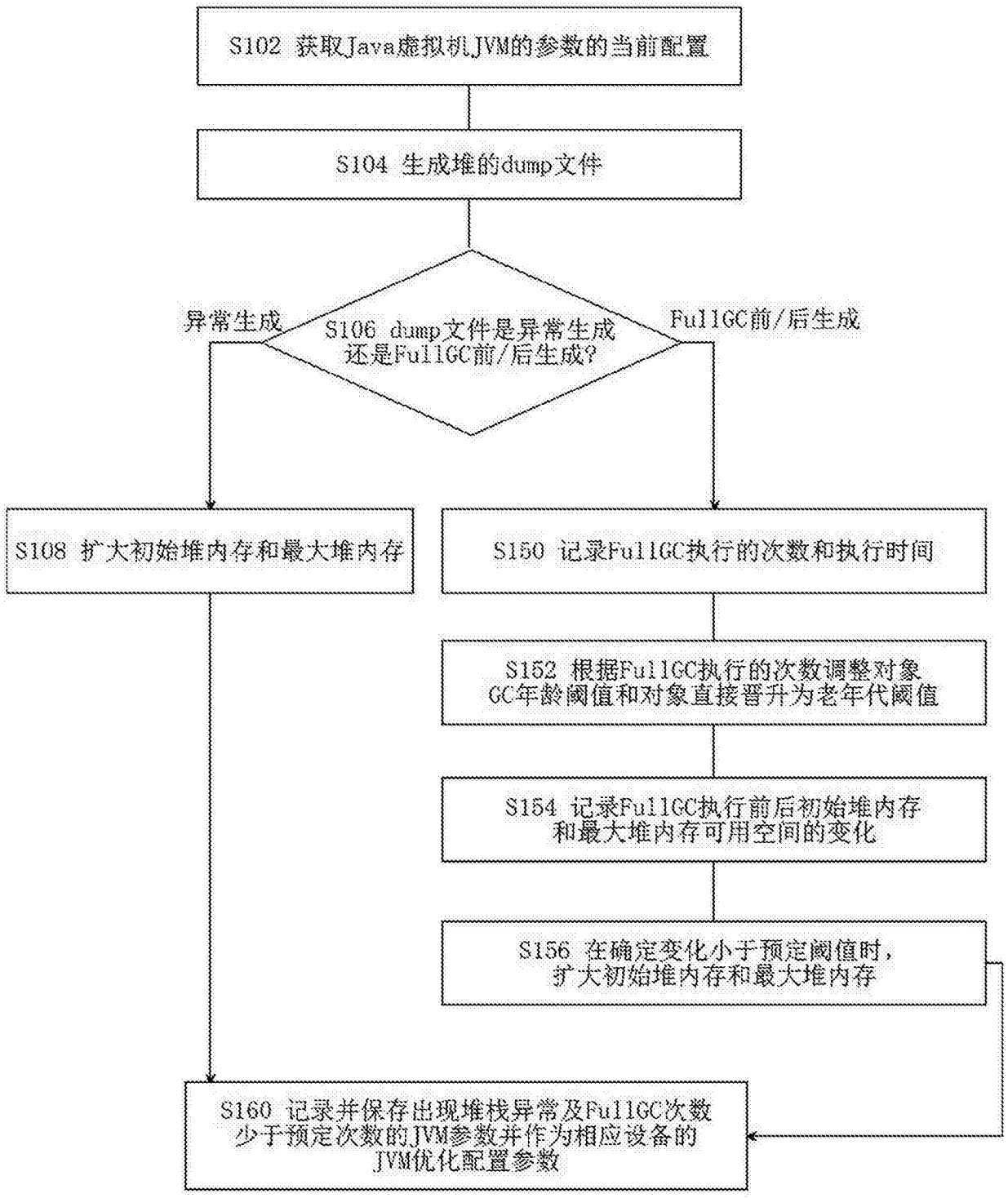

S102 获取Java虚拟机JVM的参数的当前配置
S104 生成堆的dump文件
S106 dump文件是异常生成还是FullGC前/后生成?
异常生成
FullGC前/后生成
S108 扩大初始堆内存和最大堆内存
S150 记录FullGC执行的次数和执行时间
S152 根据FullGC执行的次数调整对象GC年龄阈值和对象直接晋升为老年代阈值
S154 记录FullGC执行前后初始堆内存和最大堆内存可用空间的变化
S156 在确定变化小于预定阈值时，扩大初始堆内存和最大堆内存
S160 记录并保存出现堆栈异常及FullGC次数少于预定次数的JVM参数并作为相应设备的JVM优化配置参数

图1

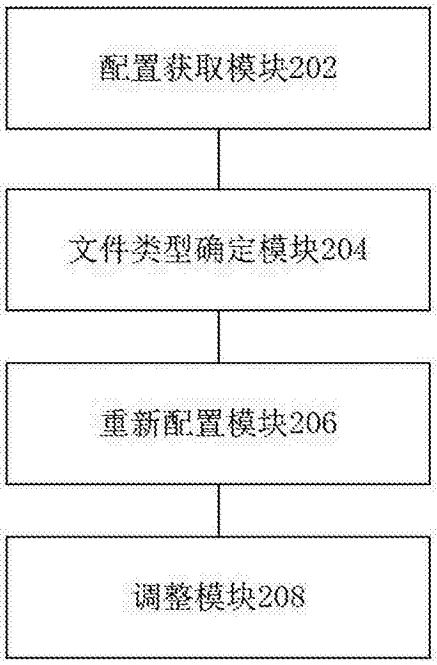


图2