



(12) 发明专利

(10) 授权公告号 CN 111045779 B

(45) 授权公告日 2024.03.01

(21) 申请号 201911058971.0

(22) 申请日 2019.11.01

(65) 同一申请的已公布的文献号

申请公布号 CN 111045779 A

(43) 申请公布日 2020.04.21

(73) 专利权人 福建天泉教育科技有限公司

地址 350212 福建省福州市长乐区湖南镇
大鹤村(福建天棣动漫科技有限公司)

(72) 发明人 刘德建 林伟 郭玉湖 陈宏

(74) 专利代理机构 福州市博深专利事务所(普

通合伙) 35214

专利代理师 张明 石晋龙

(51) Int. Cl.

G06F 9/455 (2006.01)

G06F 9/50 (2006.01)

(56) 对比文件

CN 102739680 A, 2012.10.17

CN 103617023 A, 2014.03.05

CN 105701025 A, 2016.06.22

US 2005198620 A1, 2005.09.08

审查员 王为鑫

权利要求书1页 说明书5页 附图1页

(54) 发明名称

系统内存回收配置方法、存储介质

(57) 摘要

本发明提供系统内存回收配置方法、存储介质,方法包括:依据各个虚拟机实例一天内的内存占用情况,获取各个虚拟机实例一天中对应系统繁忙时段的第一GC触发时间间隔;依据所述第一GC触发时间间隔和所述系统繁忙时段,将GC触发配置表中各个虚拟机实例对应系统繁忙时段的GC触发时间点设置为进入系统繁忙时段之时及向前推所述第一GC触发时间间隔的时间点。本发明能够灵活地配置GC操作的触发时机,通过调整系统繁忙时段的GC操作时间,实现既能保证及时内存垃圾回收,空出足够内存;又能确保业务繁忙时段系统性能的正常运行和良好的系统体验。

依据各个虚拟机实例一天内的内存占用情况,获取各个虚拟机实例一天中对应系统繁忙时段的第一GC触发时间间隔

依据所述第一GC触发时间间隔和所述系统繁忙时段,将GC触发配置表中各个虚拟机实例对应系统繁忙时段的GC触发时间点设置为进入系统繁忙时段之时及向前推所述第一GC触发时间间隔的时间点

1. 系统内存回收配置方法,其特征在于,包括:

依据各个虚拟机实例一天内的内存占用情况,获取各个虚拟机实例一天中对应系统繁忙时段的第一GC触发时间间隔;

依据所述第一GC触发时间间隔和所述系统繁忙时段,将GC触发配置表中各个虚拟机实例对应系统繁忙时段的GC触发时间点设置为进入系统繁忙时段之时及向前推所述第一GC触发时间间隔的时间点;

所述依据各个虚拟机实例一天内的内存占用情况,获取各个虚拟机实例一天中对应系统繁忙时段的GC触发时间间隔,具体为:

通过JAVA内存监控程序周期获取虚拟机实例的内存占用情况;

依据所述虚拟机实例的内存占用情况,统计各个虚拟机实例一天内的内存占用情况;

依据所述各个虚拟机实例一天内的内存占用情况,获取各个虚拟机实例一天中对应系统繁忙时段的第一GC触发时间间隔和对应系统空闲时段的第二GC触发时间间隔;

所述系统繁忙时段的获取方式有一定的策略选择,包括第一种策略:监控每时间段主要接口的调用情况,根据接口调用次数达判断是否进入繁忙时间段;第二种策略:监控业务数据量,根据下单量的数据来判断是否达到繁忙时间段;第三种策略:数据库tps的监控,根据数据库操作量判断是否达到繁忙时间段;第四种策略:缓存的获取以及设置量的监控,根据缓存量判断是否达到繁忙时间段;当上述四种策略中满足三种策略时,则说明达到了繁忙时间段;

还包括:

一异步线程定时扫描所述GC触发配置表,判断当前是否存在虚拟机实例需要触发GC;若需要,则调用对应虚拟机实例的自动触发GC接口。

2. 如权利要求1所述的系统内存回收配置方法,其特征在于,还包括:

调用所述自动触发GC接口后,将由其内部调用虚拟机实例中提供的system.gc()方法执行GC操作。

3. 如权利要求1所述的系统内存回收配置方法,其特征在于,所述GC触发配置表包括主键数据库ID字段、触发时间点字段、触发虚拟机实例ID字段、虚拟机实例的请求域名和端口号,以及是否启动状态。

4. 一种计算机可读存储介质,其上存储有计算机程序,其特征在于,所述程序在被处理器执行时,能够实现上述权利要求1-3任意一项所述的系统内存回收配置方法所包含的步骤。

系统内存回收配置方法、存储介质

技术领域

[0001] 本发明涉及JAVA虚拟机领域,具体涉及系统内存回收配置方法、存储介质。

背景技术

[0002] 现在很多的大中型软件经常使用JAVA语言来进行系统的代码开发。其具有简单、安全、稳定、跨平台的特点。例如BS结构的ERP系统、金融系统、电子商务系统、网站等几乎所有的银行系统和银行网站都是基于Java平台的。同时JAVA支持跨平台性,虽然是很好的优势,可以让JAVA在任何的系统中运行,但是前提是该系统要安装JAVA虚拟机。因此所有使用JAVA语言的系统中都需要安装JAVA虚拟机来进行语言的转化。

[0003] JAVA虚拟机在系统运行中,会针对系统使用内存所产生的各种无用数据或者无用对象进行回收处理,就是所谓的垃圾回收机制。其通过一定的策略,进行垃圾回收。包括内存达到上限,或者可使用内存不足,或者定时垃圾回收等,都是为了清理垃圾内存,回收内存。一般这样的操作都是有周期性的,一次的垃圾回收称为full GC。在进行full GC过程中,整个系统将处于卡顿状态,无法进行其他的任何操作,业务功能操作也会被挂起。一般性的系统可以接受这种短时间内的挂起操作,但如果是重要系统,或者重要节假日引起的流量洪峰时刻等,这样的full GC时刻,将较为致命,将会给系统体验上造成不好的影响。

[0004] 因此,迫切需要能够有效解决上述问题的方案,以改进系统体验。

发明内容

[0005] 本发明所要解决的技术问题是:提供系统内存回收配置方法、存储介质,确保业务繁忙阶段拥有良好的系统体验。

[0006] 为了解决上述技术问题,本发明采用的技术方案为:

[0007] 系统内存回收配置方法,包括:

[0008] 依据各个虚拟机实例一天内的内存占用情况,获取各个虚拟机实例一天中对应系统繁忙时段的第一GC触发时间间隔;

[0009] 依据所述第一GC触发时间间隔和所述系统繁忙时段,将GC触发配置表中各个虚拟机实例对应系统繁忙时段的GC触发时间点设置为进入系统繁忙时段之时及向前推所述第一GC触发时间间隔的时间点。

[0010] 本发明提供的另一个技术方案为:

[0011] 一种计算机可读存储介质,其上存储有计算机程序,所述程序在被处理器执行时,能够实现上述系统内存回收配置方法所包含的步骤。

[0012] 本发明的有益效果在于:通过统计获取各个虚拟机实例一天内在系统繁忙时段的GC触发时间间隔,并据此通过后台的GC触发配置表对各个虚拟机实例在繁忙时段的GC触发时间点进行调整,将繁忙时段内的GC触发时间点往前推,在进入繁忙时段之时做最后一次的GC。由此,既能保证系统繁忙时段的内存够用,又能大大显著提升系统繁忙时段的性能,确保在系统繁忙之时也具备良好的用户系统体验。

附图说明

[0013] 图1为本发明一实施例系统内存回收配置方法的流程示意图；

[0014] 图2为本发明实施例一的系统内存回收配置方法的流程示意图。

具体实施方式

[0015] 为详细说明本发明的技术内容、所实现目的及效果,以下结合实施方式并配合附图予以说明。

[0016] 本发明最关键的构思在于:通过后台的GC触发配置表对各个虚拟机实例在繁忙时段的GC触发时间点进行调整,将繁忙时段内的GC触发时间点往前推,在进入繁忙时段之时做最后一次的GC。

[0017] 请参照图1以及图2,本发明提供系统内存回收配置方法,包括:

[0018] 依据各个虚拟机实例一天内的内存占用情况,获取各个虚拟机实例一天中对应系统繁忙时段的第一GC触发时间间隔;

[0019] 依据所述第一GC触发时间间隔和所述系统繁忙时段,将GC触发配置表中各个虚拟机实例对应系统繁忙时段的GC触发时间点设置为进入系统繁忙时段之时及向前推所述第一GC触发时间间隔的时间点。

[0020] 从上述描述可知,本发明的有益效果在于:通过监控手段统计系统不同时间段的内存占用情况以及占用率,通过分析可以很容易的知道系统full GC的时间间隔;同时,与系统业务处理时段结合起来,在进入业务繁忙阶段之前,进行智能化的full GC触发来清理java堆中的内存垃圾,这样就可以在业务繁忙阶段降低GC触发的频率以及时间,同时具备足够内容保证系统性能运行正常,进而提高系统的效率与体验。

[0021] 进一步地,还包括:

[0022] 一异步线程定时扫描所述GC触发配置表,判断当前是否存在虚拟机实例需要触发GC;

[0023] 若需要,则调用对应虚拟机实例的自动触发GC接口。

[0024] 由上述描述可知,以异步线程定时扫描GC触发配置表的形式来实现智能化触发GC,能同时确保不对系统当前业务运行造成影响。

[0025] 进一步地,还包括:

[0026] 调用所述自动触发GC接口后,将由其内部调用虚拟机实例中提供的system.gc()方法执行GC操作。

[0027] 由上述描述可知,通过提供一个自动触发GC的接口供外部调用内部的公共的GC操作,实现外部可以自行控制GC操作。

[0028] 进一步地,所述依据各个虚拟机实例一天内的内存占用情况,获取各个虚拟机实例一天中对应系统繁忙时段的GC触发时间间隔,具体为:

[0029] 通过JAVA内存监控程序周期获取虚拟机实例的内存占用情况;

[0030] 依据所述虚拟机实例的内存占用情况,统计各个虚拟机实例一天内的内存占用情况;

[0031] 依据所述各个虚拟机实例一天内的内存占用情况,获取各个虚拟机实例一天中对应系统繁忙时段的第一GC触发时间间隔和对应系统空闲时段的第二GC触发时间间隔。

[0032] 由上述描述可知,通过监控程序调用虚拟机实例提供的数据接口,实现便捷性地获取虚拟机实例的内存占用情况,进而轻松统计获取各个虚拟机实例对应空闲时段和繁忙时段的GC触发时间间隔,为后续GC配置提供技术支持。

[0033] 进一步地,所述GC触发配置表包括主键数据库ID字段、触发时间点字段、触发虚拟机实例ID字段、虚拟机实例的请求域名和端口号,以及是否启动状态。

[0034] 由上述描述可知,通过GC触发配置表全面地记载各个虚拟机实例的各项参数信息以及对应的GC相关信息,能够保证准确无误且操作便捷地实现GC自定义触发。

[0035] 本发明提供的另一个技术方案为:

[0036] 一种计算机可读存储介质,其上存储有计算机程序,所述程序在被处理器执行时,能够实现上述系统内存回收配置方法所包含的以下步骤:

[0037] 依据各个虚拟机实例一天内的内存占用情况,获取各个虚拟机实例一天中对应系统繁忙时段的第一GC触发时间间隔;

[0038] 依据所述第一GC触发时间间隔和所述系统繁忙时段,将GC触发配置表中各个虚拟机实例对应系统繁忙时段的GC触发时间点设置为进入系统繁忙时段之时及向前推所述第一GC触发时间间隔的时间点。

[0039] 进一步地,还包括:

[0040] 一异步线程定时扫描所述GC触发配置表,判断当前是否存在虚拟机实例需要触发GC;

[0041] 若需要,则调用对应虚拟机实例的自动触发GC接口。

[0042] 进一步地,还包括:

[0043] 调用所述自动触发GC接口后,将由其内部调用虚拟机实例中提供的system.gc()方法执行GC操作。

[0044] 进一步地,所述依据各个虚拟机实例一天内的内存占用情况,获取各个虚拟机实例一天中对应系统繁忙时段的GC触发时间间隔,具体为:

[0045] 通过JAVA内存监控程序周期获取虚拟机实例的内存占用情况;

[0046] 依据所述虚拟机实例的内存占用情况,统计各个虚拟机实例一天内的内存占用情况;

[0047] 依据所述各个虚拟机实例一天内的内存占用情况,获取各个虚拟机实例一天中对应系统繁忙时段的第一GC触发时间间隔和对应系统空闲时段的第二GC触发时间间隔。

[0048] 进一步地,所述GC触发配置表包括主键数据库ID字段、触发时间点字段、触发虚拟机实例ID字段、虚拟机实例的请求域名和端口号,以及是否启动状态。

[0049] 请参照图2,本发明的实施例一为:

[0050] 提供一种系统内存回收配置方法,包括:

[0051] 首先,配置一个java内存监控程序,该程序能够每隔一段时间获取java虚拟机实例中内存的占用情况,统计出内存占用率等相关信息。该功能易于实现,因为java虚拟机中提供了查询上述数据的接口。比如说每分钟统计一次相关信息,并输出结果的曲线图以便后续使用。

[0052] 假设每个java虚拟机实例使用4G内存,监控程序每隔1分钟会获取统计一次该时刻每个java虚拟机实例的内存使用情况。这样,就可以得出一个波形图,知道每个java虚拟

机实例每分钟的内存使用情况。一般正常情况下,都是存在一个波峰波谷的波形图,则在每次波峰下降时,就说明进行了一次full GC,则每次波峰与波谷之间的时间间隔就可以很容易的理解为每次full GC所进行时间间隔了。

[0053] 此时,已经可以统计出每个java虚拟机实例每天的内存具体情况,从中可以很容易地看出每个java虚拟机实例每次GC操作的时间间隔。

[0054] 假设某一java虚拟机实例在系统空闲时段每隔1小时执行一次full GC操作;系统繁忙时段,每隔半小时执行一次full GC。系统每天的繁忙时段预计为4小时,分别为12:00-13:00,以及19:00-22:00这两个时间段。

[0055] 需要说明的是,本实施例中系统繁忙时段的获取方式有一定的策略选择。第一种策略:首先,需要提供监控程序,主要监控每时间段主要接口的调用情况。比如说,正常时间点接口调用100次,繁忙时间段该接口调用次数达到500次以上,则说明已经进入繁忙时间段。第二种策略:监控业务数据量,比如说送餐软件的下单量,根据下单量的数据来判断是否达到繁忙时间段。第三种策略:数据库tps的监控,因为繁忙时间段数据库操作量也会比平常高很多;第四种策略:缓存的获取以及设置量的监控。当上述四种策略中满足3种策略(支持自定义)时,则说明达到了繁忙时间段。

[0056] 另外,后台存在一个GC触发的配置平台,可以设置何时开始进行GC触发操作。本实施例中,对应上述获取的繁忙时段及其对应的GC触发时间间隔,将设置各个虚拟机实例对应系统繁忙时段的GC触发时间点设置为进入系统繁忙时段之时,以及进入系统繁忙时段之时向前推所述第一GC触发时间间隔的时间点。

[0057] 对应上述某一java虚拟机实例,将在每天的11:30、12:00以及18:30和19:00这4个时间点进行自动触发GC操作来回收可用内存。配置平台配置GC触发的时间点信息至配置表中,GC触发配置表存储在数据库中。GC触发配置表中存在:主键ID字段、触发时间点字段、触发JAVA实例ID以及对应JAVA实例的请求域名与端口号,以及是否启用状态。其中的主键ID字段用于记录数据库主键,数据库每条记录需要有一个主键字段。

[0058] 与此同时,每个虚拟机实例程序中,还需要提供一个自动触发GC的接口,供外部调用。该接口主要功能为调用JAVA提供的System.gc()方法来进行智能触发GC。

[0059] JAVA中提供的system.gc()为公共的垃圾收集调用方法,通过上述自动触发GC的接口,额外提供外部调用的一种使用方式,通过该种设置方式,外部可以通过调用该接口,再由该接口内部调用system.gc()方法,最终来达到外部可以进行java虚拟机实例的垃圾收集操作。

[0060] 最后,存在一个异步定时任务,每隔一分钟(支持自定义)扫描GC触发配置表,当发现需要进行GC触发操作时,调用对应JAVA实例中的自动触发GC接口来主动进行垃圾回收操作。

[0061] 通过以上方式能够实现自定义配置GC时间点并自动进行内存垃圾回收,在业务空闲期做好垃圾回收准备工作,收集好足够的内存,从而降低业务繁忙时段的GC触发频率以及时间等优点。

[0062] 本发明的实施例二为:

[0063] 本实施例对应实施例一,提供一种计算机可读存储介质,其上存储有计算机程序,所述程序在被处理器执行时,能够实现上述实施例一所述的系统内存回收配置方法所包含

的步骤。具体的步骤内容在此不进行复述,详细请参阅实施例一的记载。

[0064] 综上所述,本发明提供的系统内存回收配置方法、存储介质,能够灵活地配置GC操作的触发时机,通过调整系统繁忙时段的GC操作时间,实现既能保证及时内存垃圾回收,空出足够内存;又能确保业务繁忙时段系统性能的正常运行和良好的系统体验;进一步地,还具备易于实现、内存占用少以及分析处理准确性高等优点。

[0065] 以上所述仅为本发明的实施例,并非因此限制本发明的专利范围,凡是利用本发明说明书及附图内容所作的等同变换,或直接或间接运用在相关的技术领域,均同理包括在本发明的专利保护范围内。

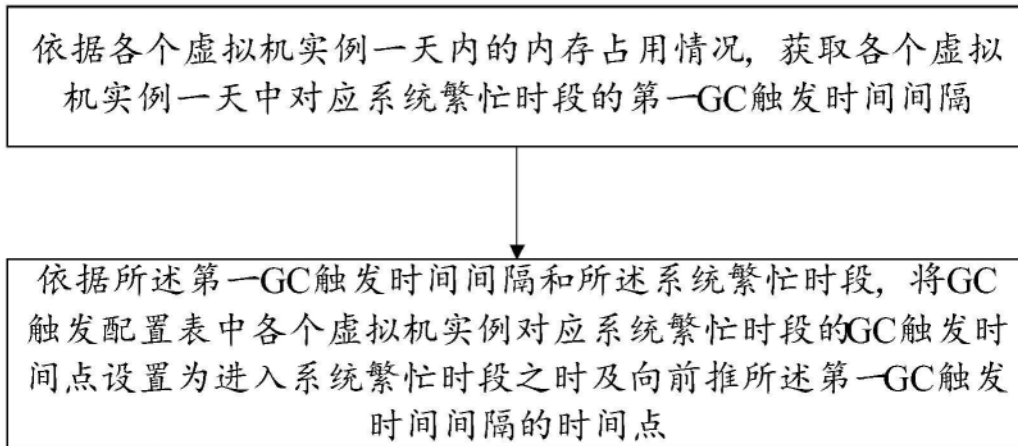


图1

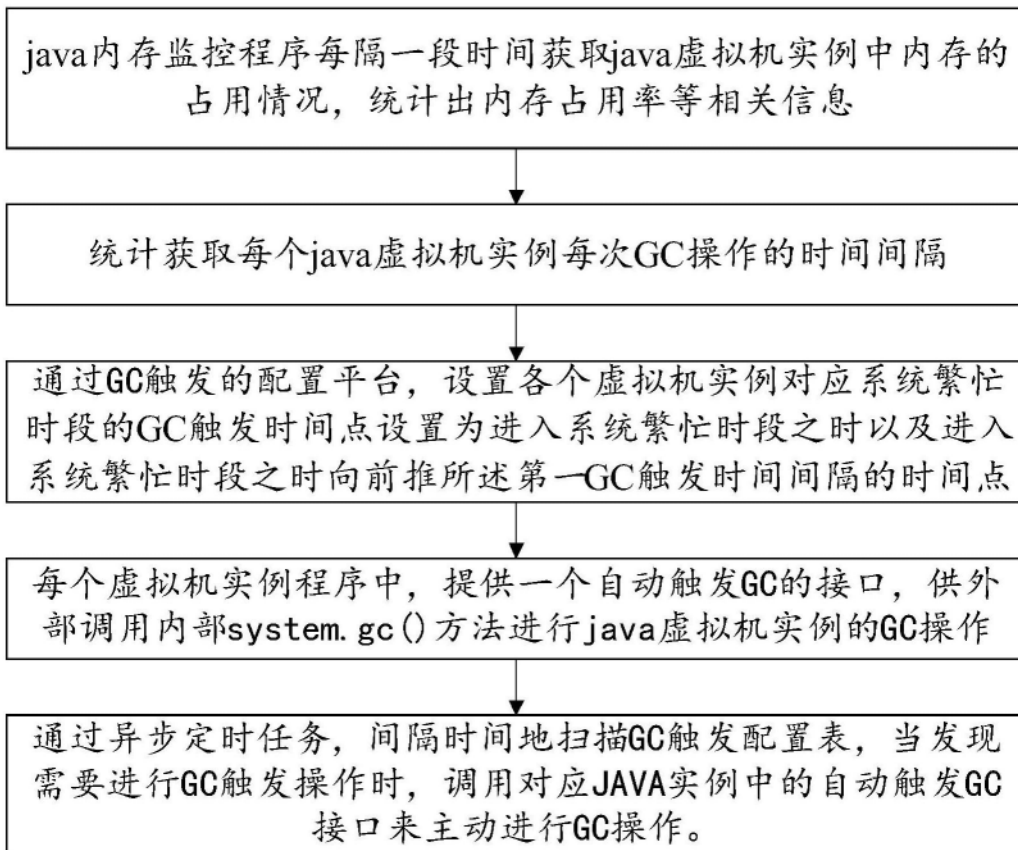


图2