



(12) 发明专利申请

(10) 申请公布号 CN 103984582 A

(43) 申请公布日 2014. 08. 13

(21) 申请号 201410245591. 9

(22) 申请日 2014. 06. 04

(71) 申请人 网易(杭州)网络有限公司

地址 310052 浙江省杭州市滨江区长河街道  
网商路 599 号 4 幢 7 层

(72) 发明人 张伟土 卢立祎

(74) 专利代理机构 北京信远达知识产权代理事

务所(普通合伙) 11304

代理人 赵百令 刘大玲

(51) Int. Cl.

G06F 9/445(2006. 01)

G06F 9/45(2006. 01)

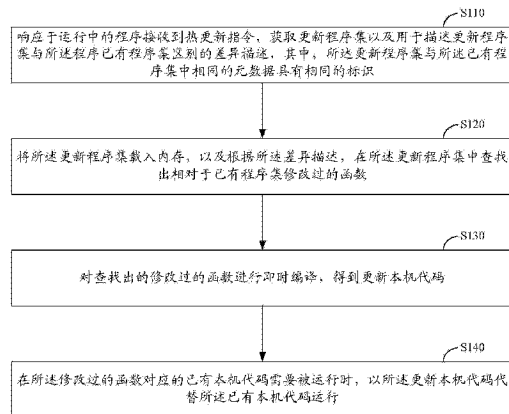
权利要求书3页 说明书7页 附图2页

(54) 发明名称

一种热更新方法和装置

(57) 摘要

本发明的实施方式提供了一种热更新方法和装置。例如,该方法可以包括:响应于运行中的程序接收到热更新指令,获取更新程序集以及用于描述更新程序集与程序已有程序集区别的差异描述,其中,更新程序集与已有程序集中相同的元数据具有相同的标识;将更新程序集载入内存,以及根据差异描述,在更新程序集中查找出相对于已有程序集修改过的函数;对查找出的修改过的函数进行即时编译得到更新本机代码;在修改过的函数对应的已有本机代码需要被运行时,以更新本机代码代替已有本机代码运行。由于更新程序集与已有程序集相同的元数据具有相同的标识,更新后已有对象仍然可用,对程序设计没有特殊要求。此外,本发明的实施方式提供了一种热更新装置。



1. 一种热更新方法,包括:

响应于运行中的程序接收到热更新指令,获取更新程序集以及用于描述更新程序集与  
所述程序已有程序集区别的差异描述,其中,所述更新程序集与所述已有程序集中相同的  
元数据具有相同的标识;

将所述更新程序集载入内存,以及根据所述差异描述,在所述更新程序集中查找出相  
对于已有程序集修改过的函数;

对查找出的修改过的函数进行即时编译,得到更新本机代码;

在所述修改过的函数对应的已有本机代码需要被运行时,以所述更新本机代码代替所  
述已有本机代码运行。

2. 根据权利要求 1 所述的方法,在响应于运行中的程序接收到热更新指令,获取更新  
程序集以及用于描述更新程序集与程序已有程序集区别的差异描述之前,还包括:

对根据热更新需求修改的源代码进行编译,生成更新程序集;

对所述更新程序集进行结构调整,使调整后的更新程序集与所述程序的已有程序集中  
相同的元数据具有相同的标识;

将所述更新程序集与所述已有程序集进行比较,生成差异描述文件,其中,所述差异描  
述文件包含用于描述更新程序集与已有程序集区别的差异描述;

向所述运行中的程序发送热更新指令,以使所述运行中的程序从所述差异描述文件  
中获取所述差异描述。

3. 根据权利要求 1 所述的方法,还包括:将所述修改过的函数对应的已有本机代码的  
第一条指令修改为跳转到所述更新本机代码处运行的跳转指令;

所述在修改过的函数对应的已有本机代码需要被运行时,以更新本机代码代替已有本  
机代码运行的具体实现为:在修改过的函数对应的已有本机代码需要被运行时,进入所述  
已有本机代码的第一条指令运行,从而由所述跳转指令的运行跳转到所述更新本机代码处  
运行。

4. 根据权利要求 3 所述的方法,其中,如果所述修改过的函数为泛型函数,进一步查找  
出所有使用所述泛型函数实例化出的函数;

针对每个使用所述泛型函数实例化出的函数,分别进行即时编译,得到对应的更新本  
机代码;

将每个使用所述泛型函数实例化出的函数对应的已有本机代码的第一条指令修改为  
跳转到对应的所述更新本机代码处运行的跳转指令。

5. 根据权利要求 1 所述的方法,还包括:

如果根据所述差异描述确定需要对指定方法的访问级别和 / 或指定字段的访问级别  
进行修改,查找出所述指定方法对应的对象和 / 或指定字段对应的对象;

对查找出的对象的访问属性进行相应修改。

6. 根据权利要求 1 所述的方法,其中,所述更新程序集中包括:用于在全局哈希表中记  
录增加的字段与已有对象的映射关系的特定函数;

在该特定函数对应的本机代码运行时,所述本机代码执行在所述全局哈希表中记录所  
述增加的字段与已有对象的映射关系的步骤;

在对所述已有对象的引用期间,根据所述全局哈希表记录的映射关系对所述已有对象

对应的增加的字段进行访问。

7. 根据权利要求 6 所述的方法,其中,在所述全局哈希表中记录映射关系而产生的对所述已有对象的引用为弱引用;

且,还包括:按需对所述全局哈希表中记录的映射关系进行遍历,在遍历的过程中,对遍历到的映射关系中对应的对象是否被引用进行检查,如果否,则将该未被引用的对象的映射关系从所述全局哈希表中删除。

8. 根据权利要求 2 所述的方法,在对所述更新程序集进行结构调整之前,还包括:

如果所述更新程序集以及已有程序集中由编译器优化而产生的类定义不同,将所述更新程序集中由编译器优化而产生的类的名称进行更改,以及将已有程序集中由编译器优化而产生的类定义复制到更新程序集中。

9. 一种热更新装置,包括:

接收单元,配置用于响应于运行中的程序接收到热更新指令,获取更新程序集以及用于描述更新程序集与所述程序已有程序集区分的差异描述,其中,所述更新程序集与所述已有程序集中相同的元数据具有相同的标识;

修改确定单元,配置用于将所述更新程序集载入内存,以及根据所述差异描述,在所述更新程序集中查找出相对于已有程序集修改过的函数;

即时编译单元,配置用于对查找出的修改过的函数进行即时编译,得到更新本机代码;

运行单元,配置用于在所述修改过的函数对应的已有本机代码需要被运行时,以所述更新本机代码代替所述已有本机代码运行。

10. 根据权利要求 9 所述的装置,还包括:

新集生成单元,配置用于对根据热更新需求修改的源代码进行编译,生成更新程序集;

结构调整单元,配置用于对所述更新程序集进行结构调整,使调整后的更新程序集与所述程序的已有程序集中相同的元数据具有相同的标识;

比较单元,配置用于将所述更新程序集与所述已有程序集进行比较,生成差异描述文件,其中,所述差异描述文件包含用于描述更新程序集与已有程序集区分的差异描述;

指令发送单元,配置用于向所述运行中的程序发送热更新指令,以使所述运行中的程序从所述差异描述文件中获取所述差异描述。

11. 根据权利要求 9 所述的装置,还包括:修改指令单元,配置用于将所述修改过的函数对应的已有本机代码的第一条指令修改为跳转到所述更新本机代码处运行的跳转指令;

其中,所述运行单元,配置用于在修改过的函数对应的已有本机代码需要被运行时,进入所述已有本机代码的第一条指令运行,从而由所述跳转指令的运行跳转到所述更新本机代码处运行。

12. 根据权利要求 11 所述的装置,还包括:泛型函数查找单元,配置用于如果所述修改过的函数为泛型函数,在所述更新程序集中查找出所有使用所述泛型函数实例化出的函数;

所述即时编译单元,配置用于针对泛型函数查找单元查找出的每个使用所述泛型函数

实例化出的函数,分别进行即时编译,得到对应的更新本机代码;

所述运行单元,配置用于将所述泛型函数查找单元查找出的每个使用所述泛型函数实例化出的函数对应的已有本机代码的第一条指令修改为跳转到对应的所述更新本机代码处运行的跳转指令。

13. 根据权利要求 9 所述的装置,还包括:

对象查找单元,配置用于如果根据所述差异描述确定需要对指定方法的访问级别和/或指定字段的访问级别进行修改,查找出所述指定方法对应的对象和/或指定字段对应的对象;

属性修改单元,配置用于对所述对象查找单元查找出的对象的访问属性进行相应修改。

14. 根据权利要求 9 所述的装置,其中,所述更新程序集中包括:用于在全局哈希表中记录增加的字段与已有对象的映射关系的特定函数;

在该特定函数对应的本机代码运行时,所述本机代码执行在所述全局哈希表中记录所述增加的字段与已有对象的映射关系的步骤;

还包括,访问单元,配置用于在对所述已有对象的引用期间,根据所述全局哈希表记录的映射关系对所述已有对象对应的增加的字段进行访问。

15. 根据权利要求 14 所述的装置,其中,在所述全局哈希表中记录映射关系而产生的对所述已有对象的引用为弱引用;

且,还包括:检查单元,配置用于按需对所述全局哈希表中记录的映射关系进行遍历,在遍历的过程中,对遍历到的映射关系中对应的对象是否被引用进行检查,如果否,则将该未被引用的对象的映射关系从所述全局哈希表中删除。

16. 根据权利要求 10 所述的装置,还包括:

优化处理单元,配置用于在所述结构调整单元对所述更新程序集进行结构调整之前,如果所述更新程序集以及已有程序集中由编译器优化而产生的类定义不同,将所述更新程序集中由编译器优化而产生的类的名称进行更改,以及将已有程序集中由编译器优化而产生的类定义复制到更新程序集中。

## 一种热更新方法和装置

### 技术领域

[0001] 本发明的实施方式涉及计算机应用领域,更具体地,本发明的实施方式涉及一种热更新方法和装置。

### 背景技术

[0002] 本部分旨在为权利要求书中陈述的本发明的实施方式提供背景或上下文。此处的描述不因为包括在本部分中就承认是现有技术。

[0003] 对于通过互联网提供服务的系统来说,当系统运行期间发现急需改正的问题时,需要及时修复以保证服务质量。为了使用户有更好的体验,一般采用在线热更新的方式修复问题。

[0004] 目前,热更新普遍的做法是,在系统开发时,将方法(包含一系列语句的代码块)模块化,将各个模块设计成单独的DLL(Dynamic Link Library,动态链接库),其中,要求各个模块不能通过变量记录运行状态,否则一旦卸载原有的DLL将导致变量所记录的运行状态丢失;要求类定义与方法定义分离,否则将导致原有模块中的类定义所产生的对象在原有的DLL卸载后不可用。这样,通过上述做法实现的系统可以通过动态卸载原有的DLL加载更新的DLL实现热更新。

### 发明内容

[0005] 但是,由于现有技术动态卸/加载DLL的方式热更新,对系统程序设计提出了特定要求,因此在现有技术中,如何热更新仍然是非常令人烦恼的问题。

[0006] 为此,非常需要一种改进的热更新方法,以实现系统程序设计不存在特定要求的目的。

[0007] 在本上下文中,本发明的实施方式期望提供一种热更新方法及装置。

[0008] 在本发明实施方式的第一方面中,提供了一种热更新方法,例如,该方法包括:响应于运行中的程序接收到热更新指令,获取更新程序集以及用于描述更新程序集与所述程序已有程序集区别的差异描述,其中,所述更新程序集与所述已有程序集中相同的元数据具有相同的标识;将所述更新程序集载入内存,以及根据所述差异描述,在所述更新程序集中查找出相对于已有程序集修改过的函数;对查找出的修改过的函数进行即时编译,得到更新本机代码;在所述修改过的函数对应的已有本机代码需要被运行时,以所述更新本机代码代替所述已有本机代码运行。

[0009] 在本发明实施方式的第二方面中,提供了一种热更新装置,例如,该装置可以包括:接收单元,可以配置用于响应于运行中的程序接收到热更新指令,获取更新程序集以及用于描述更新程序集与所述程序已有程序集区别的差异描述,其中,所述更新程序集与所述已有程序集中相同的元数据具有相同的标识;修改确定单元,可以配置用于将所述更新程序集载入内存,以及根据所述差异描述,在所述更新程序集中查找出相对于已有程序集修改过的函数;即时编译单元,可以配置用于对查找出的修改过的函数进行即时编译,得到

更新本机代码；运行单元，可以配置用于在所述修改过的函数对应的已有本机代码需要被运行时，以所述更新本机代码代替所述已有本机代码运行。

[0010] 根据本发明实施方式的热更新方法和装置，由于更新程序集与程序已有程序集相同的元数据具有相同的标识，更新程序集中相对于系统已有程序集修改过的函数进行即时编译得到的更新本机代码仍然可以使用已有对象，因此，本发明方法以更新本机代码代替修改过的函数对应的已有本机代码运行就可以使程序在运行过程中被更新，对程序的源代码设计没有特定要求。

#### 附图说明

[0011] 通过参考附图阅读下文的详细描述，本发明示例性实施方式的上述以及其他目的、特征和优点将变得易于理解。在附图中，以示例性而非限制性的方式示出了本发明的若干实施方式，其中：

[0012] 图 1 示意性地示出了根据本发明实施方式的热更新方法流程示意图；

[0013] 图 2 示意性地示出了根据本发明一实施例的差异描述文件示意图；

[0014] 图 3 示意性地示出了根据本发明实施方式的热更新装置结构示意图；

[0015] 在附图中，相同或对应的标号表示相同或对应的部分。

#### 具体实施方式

[0016] 下面将参考若干示例性实施方式来描述本发明的原理和精神。应当理解，给出这些实施方式仅仅是为了使本领域技术人员能够更好地理解进而实现本发明，而并非以任何方式限制本发明的范围。相反，提供这些实施方式是为了使本公开更加透彻和完整，并且能够将本公开的范围完整地传达给本领域的技术人员。

[0017] 本领域技术人员知道，本发明的实施方式可以实现为一种系统、装置、设备、方法或计算机程序产品。因此，本公开可以具体实现为以下形式，即：完全的硬件、完全的软件（包括固件、驻留软件、微代码等），或者硬件和软件结合的形式。

[0018] 根据本发明的实施方式，提出了一种热更新方法和装置。

[0019] 在本文中，需要理解的是，附图中的任何元素数量均用于示例而非限制，以及任何命名都仅用于区分，而不具有任何限制含义。

[0020] 下面参考本发明的若干代表性实施方式，详细阐释本发明的原理和精神。

#### [0021] 发明概述

[0022] 元数据是存储在程序集中用于描述程序（如类型和方法或称函数）的数据。运行时库通过元数据的标识来引用元数据以发现类、成员、继承等信息。本发明人发现，对于根据更新需求修改源代码编译得到的更新程序集来说，如果更新程序集与系统已有程序集相同的元数据具有相同的标识（或称，具有相同的编号），那么已有对象仍然有效，所以更新程序集中相对于系统已有程序集修改过的函数进行即时编译得到的更新本机代码仍然可以使用已有对象，系统运行中记录动态的数据不会丢失，因此，以更新本机代码代替修改过的函数对应的已有本机代码运行就可以使系统在运行过程中被更新，对系统程序的源代码设计没有特定要求。

[0023] 在介绍了本发明的基本原理之后，下面具体介绍本发明的各种非限制性实施方

式。

[0024] 应用场景总览

[0025] 例如,本发明示例性实施方式的热更新方法可以应用于对大型网络游戏系统(包括服务器端和/或客户端)的在线更新。

[0026] 示例性方法

[0027] 下面结合上述应用场景,参考图1来描述根据本发明示例性实施方式的热更新方法。需要注意的是,上述应用场景仅是为了便于理解本发明的精神和原理而示出,本发明的实施方式在此方面不受任何限制。相反,本发明的实施方式可以应用于适用的任何场景。

[0028] 例如,参见图1,为本发明实施例提供的热更新方法流程示意图。如图所示,本发明实施例一种可能的实施方式可以包括:

[0029] S110、响应于运行中的程序接收到热更新指令,获取更新程序集以及用于描述更新程序集与所述程序已有程序集区分的差异描述,其中,所述更新程序集与所述已有程序集中相同的元数据具有相同的标识。

[0030] 其中,在运行中的程序接收到热更新指令之前,还可以对根据热更新需求修改的源代码进行编译,生成更新程序集;对所述更新程序集进行结构调整,使调整后的更新程序集与所述程序的已有程序集中相同的元数据具有相同的标识;将所述更新程序集与所述已有程序集进行比较,生成差异描述文件,其中,所述差异描述文件包含用于描述更新程序集与已有程序集区分的差异描述;向所述运行中的程序发送热更新指令,以使所述运行中的程序从所述差异描述文件中获取所述差异描述。例如,可以向运行中的程序发送热更新指令以及差异描述文件,在运行中的程序对差异描述文件进行解析确定差异描述文件对应的热更新程序集之后,运行中的程序相应调用运行环境提供的热更新接口,进而在运行环境(例如, Mono,跨平台的 .NET 运行环境)层面读取该更新程序集并执行后续热更新操作。为了减少需要传输的数据量,例如,可以利用用于比较二进制文件差异的 xdelta 工具比较更新程序集与已有程序集(其中,更新程序集与已有程序集相同元数据具有相同标识),生成仅包含更新程序集相对于已有程序集不同部分的二进制 delta 数据,将该生成的二进制 delta 数据作为更新程序集发送给运行中的程序。例如,根据热更新需求修改的源代码可以为各种编程语言的源代码,例如, C#、Visual Basic.NET、C++/CLI Managed、F#、J#、JScript.NET、Windows PowerShell 等,作出的修改可以包括添加新类、修改函数、改变函数或字段的访问级别、为已有对象增加字段、等等。源代码编译得到的更新程序集可以是人类可读的通用中间语言 CIL。更新程序集可以以 dll 或者 exe 文件的形式存在。

[0031] 需要说明的是,由于需要将更新程序集调整为与已有程序集中相同的元数据具有相同的标识以便在更新之后已有对象仍然可用,因此,在对更新程序集进行与已有程序集结构相同的调整时,调整后的更新程序集元数据中的 String Heap(字符串堆)、User String Heap(用户字符串堆)、Blob Heap(二进制对象堆)只能在已有程序集相同元数据原有内容基础上增加内容,而不能修改原有内容。其中,String Heap 被元数据引用,用于表示方法名或称函数名、字段名、类名、变量名以及资源相关字符串等;User String Heap 包含定义在源代码中的字符串;Blob Heap 包含元数据引用的二进制对象。

[0032] 根据本发明一些可能的实施方式,可以利用 differ 工具(一种比较工具)将更新程序集与所述已有程序集进行逐类、逐函数、逐语句的比较。在比较的同时,可以对更新程

序集进行与已有程序集结构相同的调整。其中,差异描述文件可以包含更新程序集与已有程序集所有区别的差异描述,如类被修改的描述、函数被修改的描述、字段的属性被修改的描述等等。在一些可能的实施方式中,本发明还对是否存在超出热更新能力的修改(例如,删除已有类定义为超出热更新能力的修改)进行监测,如果存在,则返回错误提示,如果不存在,正常生成差异描述文件。例如,差异描述文件中不同的差异描述可以按照片段划分,每个片段按照其要表达的含义根据预设的编码规则进行编码。例如,差异描述可以为图2所示的内容。为了便于理解,图2中不同片段用不同底色表示。参见图2,如片段“EE 11 0000 00 43 6C 61 73 73 4C 69 62 72 61 72 79 31 2E 64 6C 6C”表达的含义可以是更新程序集的名称“ClassLibrary1.dll”;如片段“83 02 00 00 06”、“8303 00 00 06”、“83 06 00 00 06”、“83 07 00 00 06”、“83 09 00 00 06”、以及“83 12 00 00 06”分别表达的含义可以是ID分别为06000002、06000003、06000006、06000007、06000009、06000012这6个方法或称函数被改变;如片段“85 08 00 00 04 06 00”、“85 09 00 00 04 06 00”、“850A 00 00 04 01 00”分别表达的含义可以是三个字段的属性被改变。

[0033] S120、将所述更新程序集载入内存,以及根据所述差异描述,在所述更新程序集中查找出相对于已有程序集修改过的函数。

[0034] 例如,根据图2所示的差异描述,可以确定ID分别为06000002、06000003、06000006、06000007、06000009、06000012这6个函数的函数体被改变,在更新程序集中查找出ID分别为06000002、06000003、06000006、06000007、06000009、06000012这6个函数的函数体。

[0035] 需要说明的是,在更新程序集载入内存时,已有程序集可保留在内存中或者也可以去除,具体可以根据已有程序集是否存在仍然需要被利用的数据决定。

[0036] S130、对查找出的修改过的函数进行即时编译,得到更新本机代码。

[0037] 即时编译(JIT, Just-in-time Compiler),是指在运行时把中间语言如CIL编译成本机代码。本机代码(native code),是指被编译为特定处理器机器码的代码,可被处理器直接执行。

[0038] S140、在所述修改过的函数对应的已有本机代码需要被运行时,以所述更新本机代码代替所述已有本机代码运行。

[0039] 例如,可以将修改过的函数对应的已有本机代码的第一条指令修改为跳转到所述更新本机代码处运行的跳转指令,在修改过的函数对应的已有本机代码需要被运行时,进入所述已有本机代码的第一条指令运行,从而由所述跳转指令的运行跳转到所述更新本机代码处运行。为了找到已有本机代码在内存中的位置修改第一条指令,例如,可以在已有程序集中的函数被编译为本机代码时,记录下函数的ID与该函数对应的已有本机代码在内存中的位置的对应关系,从而在需要修改已有本机代码的第一条指令时,根据该对应关系找到修改过的函数对应的已有本机代码在内存中的位置,对该位置的第一条指令进行修改。

[0040] 根据本发明一些可能的实施方式,如果所述修改过的函数为泛型函数,则还需进一步查找出所有使用所述泛型函数实例化出的函数;针对每个使用所述泛型函数实例化出的函数,分别进行即时编译,得到对应的更新本机代码;将每个使用所述泛型函数实例化出的函数对应的已有本机代码的第一条指令修改为跳转到对应的所述更新本机代码处运行



的跳转指令。例如,可以在泛型函数实例化出函数时,将泛型函数与其实例化出的函数的映射关系记录下来,例如可以记录在哈希表中,如果修改过的函数为泛型函数,则可以在泛型函数与其实例化出的函数的映射关系记录中查找出所有使用所述泛型函数实例化出的函数。

[0041] 根据本发明一些可能的实施方式,如果根据所述差异描述确定需要对指定方法的访问级别和 / 或指定字段的访问级别进行修改,则可以查找出所述指定方法对应的对象和 / 或指定字段对应的对象;对查找出的对象的访问属性进行相应修改。例如,该实施方式可以将方法或字段的访问级别从 public 改为 private,反之亦然。

[0042] 根据本发明一些可能的实施方式,还可以为已有对象增加字段。所述字段可以为任何类型的变量。例如,其具体实施方式可以为:使所述更新程序集中包含用于在全局哈希表中记录增加的字段与已有对象的映射关系的特定函数;在该特定函数对应的本机代码运行时,所述本机代码执行在所述全局哈希表中记录所述增加的字段与已有对象的映射关系的步骤;在对所述已有对象的引用期间,根据所述全局哈希表记录的映射关系对所述已有对象对应的增加的字段进行访问。在该实施方式中,全局哈希表 key-value 的映射关系中, key 可以是已有对象, value 可以相当于是该已有对象新增的字段的容器。另外,为了能够及时释放不被引用的对象所占用的内存,在所述全局哈希表中记录映射关系而产生的对所述已有对象的引用可以为弱引用,从而当已有对象未被引用时,垃圾回收器可及时回收该对象,释放其占用的内存。另外,为了提高全局哈希表的查询效率,还可以按需对所述全局哈希表中记录的映射关系进行遍历,在遍历的过程中,对遍历到的映射关系中对应的对象是否被引用进行检查,如果否,则将该未被引用的对象的映射关系从所述全局哈希表中删除,从而使得该全局哈希表中仅保存需要使用的映射关系,提高查询映射关系的效率。

[0043] 根据本发明一些可能的实施方式,更新程序集和已有程序集中可能存在编译器编译源代码时为了优化而产生的类。更新程序集和已有程序集中为优化而产生的类的内部字段和内部类数量可能不同。可以理解的是,程序集中的元数据的标识例如编号按照元数据的数量顺序编排,如果更新程序集和已有程序集中为了优化而产生的类的内部字段和内部类数量不同,将导致更新程序集与已有程序集中某些相同元数据将被编排到不同的标识,进而导致某些已有对象不能使用。因此,为了使得更新程序集与已有程序集相同的元数据能够标识一致,在对所述更新程序集进行结构调整之前,本发明实施例还对更新程序集和已有程序集中由编译器优化而产生的类定义进行比较,如果所述更新程序集以及已有程序集中由编译器优化而产生的类定义不同,将所述更新程序集中由编译器优化而产生的类的名称进行更改,以及将已有程序集中由编译器优化而产生的类定义复制到更新程序集中。例如,C# 编译器由于优化而产生的类为 <PrivateImplementationDetails>,更新程序集中的由编译器优化而产生的类的名称可以改为 <PrivateImplementationDetails>{MVID},由于从已有程序集复制过来类 <PrivateImplementationDetails>,因此更新后可以保持数据的一致性,使所有已有对象仍然可用。

[0044] 可见,应用本发明实施例提供的热更新方法,由于所接收的更新程序集与已有程序集中相同的元数据具有相同的标识,对修改过的函数进行即时编译得到的更新本机代码可以使用已有对象,程序运行中记录动态的数据不会丢失,因此,在修改过的函数对应的已有本机代码需要被运行时,以更新本机代码代替已有本机代码运行就可以在对源代码设计

没有特定要求的情况下进行热更新。

[0045] 示例性设备

[0046] 在介绍了本发明示例性实施方式的方法之后,接下来,参考图 3 对本发明示例性实施方式的热更新装置。

[0047] 例如,参见图 3,为本发明实施例提供的热更新装置结构示意图。如图所示,本发明实施例一种可能的实施方式可以包括:

[0048] 接收单元 310,可以配置用于响应于运行中的程序接收到热更新指令,获取更新程序集以及用于描述更新程序集与所述程序已有程序集区别的差异描述,其中,所述更新程序集与所述已有程序集中相同的元数据具有相同的标识;修改确定单元 320,可以配置用于将所述更新程序集载入内存,以及根据所述差异描述,在所述更新程序集中查找出相对于已有程序集修改过的函数;即时编译单元 330,可以配置用于对查找出的修改过的函数进行即时编译,得到更新本机代码;运行单元 340,可以配置用于在所述修改过的函数对应的已有本机代码需要被运行时,以所述更新本机代码代替所述已有本机代码运行。

[0049] 由于本发明实施例提供的装置的接收单元 310 所接收的更新程序集与已有程序集中相同的元数据具有相同的标识,即时编译单元 330 对修改确定单元 320 查找出的修改过的函数进行即时编译得到的更新本机代码可以使用已有对象,系统运行中记录动态的数据不会丢失,因此,运行单元 340 在修改过的函数对应的已有本机代码需要被运行时,以更新本机代码代替已有本机代码运行就可以在对源代码设计没有特定要求的情况下进行热更新。

[0050] 根据本发明一些可能的实施方式,该热更新装置还可以包括:新集生成单元 350,可以配置用于对根据热更新需求修改的源代码进行编译,生成更新程序集;结构调整单元 351,可以配置用于对所述更新程序集进行结构调整,使调整后的更新程序集与所述程序的已有程序集中相同的元数据具有相同的标识;比较单元 352,可以配置用于将所述更新程序集与所述已有程序集进行比较,生成差异描述文件,其中,所述差异描述文件包含用于描述更新程序集与已有程序集区别的差异描述;指令发送单元 353,可以配置用于向所述运行中的程序发送热更新指令,以使所述运行中的程序从所述差异描述文件中获取所述差异描述。

[0051] 根据本发明一些可能的实施方式,该热更新装置还可以包括:修改指令单元 360,可以配置用于将所述修改过的函数对应的已有本机代码的第一条指令修改为跳转到所述更新本机代码处运行的跳转指令。该热更新装置的所述运行单元 340,可以配置用于在修改过的函数对应的已有本机代码需要被运行时,进入所述已有本机代码的第一条指令运行,从而由所述跳转指令的运行跳转到所述更新本机代码处运行。

[0052] 根据本发明一些可能的实施方式,该热更新装置还可以包括:泛型函数查找单元 361,可以配置用于如果所述修改过的函数为泛型函数,在所述更新程序集中查找出所有使用所述泛型函数实例化出的函数。该热更新装置的所述即时编译单元 330,可以配置用于针对泛型函数查找单元查找出的每个使用所述泛型函数实例化出的函数,分别进行即时编译,得到对应的更新本机代码。所述修改指令单元 360,可以配置用于将所述泛型函数查找单元查找出的每个使用所述泛型函数实例化出的函数对应的已有本机代码的第一条指令修改为跳转到对应的所述更新本机代码处运行的跳转指令。

[0053] 根据本发明一些可能的实施方式,该热更新装置还可以包括:对象查找单元 370,可以配置用于如果根据所述差异描述确定需要对指定方法的访问级别和/或指定字段的访问级别进行修改,查找出所述指定方法对应的对象和/或指定字段对应的对象;属性修改单元 371,可以配置用于对所述对象查找单元查找出的对象的访问属性进行相应修改。

[0054] 根据本发明一些可能的实施方式,所述更新程序集中可以包括:用于在全局哈希表中记录增加的字段与已有对象的映射关系的特定函数。在该特定函数对应的本机代码运行时,所述本机代码执行在所述全局哈希表中记录所述增加的字段与已有对象的映射关系的步骤。相应地,该热更新装置还可以包括,访问单元 380,可以配置用于在对所述已有对象的引用期间,根据所述全局哈希表记录的映射关系对所述已有对象对应的增加的字段进行访问。在该实施方式的一种可能的实施方式中,在所述全局哈希表中记录映射关系而产生的对所述已有对象的引用可以为弱引用。且,还可以包括:检查单元 381,可以配置用于按需对所述全局哈希表中记录的映射关系进行遍历,在遍历的过程中,对遍历到的映射关系中所对应的对象是否被引用进行检查,如果否,则将该未被引用的对象的映射关系从所述全局哈希表中删除。

[0055] 根据本发明一些可能的实施方式,更新程序集和已有程序集中都可能存在编译器编译源代码时为了优化而产生的类。为了使得更新程序集与已有程序集相同元数据能够标识一致,该热更新装置还可以包括,优化处理单元 354,可以配置用于在所述结构调整单元 351 对所述更新程序集进行结构调整之前,如果所述更新程序集以及已有程序集中由编译器优化而产生的类定义不同,将所述更新程序集中由编译器优化而产生的类的名称进行更改,以及将已有程序集中由编译器优化而产生的类定义复制到更新程序集中。

[0056] 需要注意的是,新集生成单元 350、结构调整单元 351、比较单元 352、指令发送单元 353、修改指令单元 360、泛型函数查找单元 361、对象查找单元 370、属性修改单元 371、访问单元 380、检查单元 381、以及优化处理单元 354 在图 3 中用虚线绘制,以表示这些单元不是本发明装置的必要单元。

[0057] 应当注意,尽管在上文详细描述中提及了热更新装置的若干单元或子单元,但是这种划分仅仅并非强制性的。实际上,根据本发明的实施方式,上文描述的两个或更多单元的特征和功能可以在一个单元中具体化。反之,上文描述的一个单元的特征和功能可以进一步划分为由多个单元来具体化。

[0058] 此外,尽管在附图中以特定顺序描述了本发明方法的操作,但是,这并非要求或者暗示必须按照该特定顺序来执行这些操作,或是必须执行全部所示的操作才能实现期望的结果。附加地或备选地,可以省略某些步骤,将多个步骤合并为一个步骤执行,和/或将一个步骤分解为多个步骤执行。

[0059] 虽然已经参考若干具体实施方式描述了本发明的精神和原理,但是应该理解,本发明并不限于所公开的具体实施方式,对各方面的划分也不意味着这些方面中的特征不能组合以进行受益,这种划分仅是为了表述的方便。本发明旨在涵盖所附权利要求的精神和范围内所包括的各种修改和等同布置。

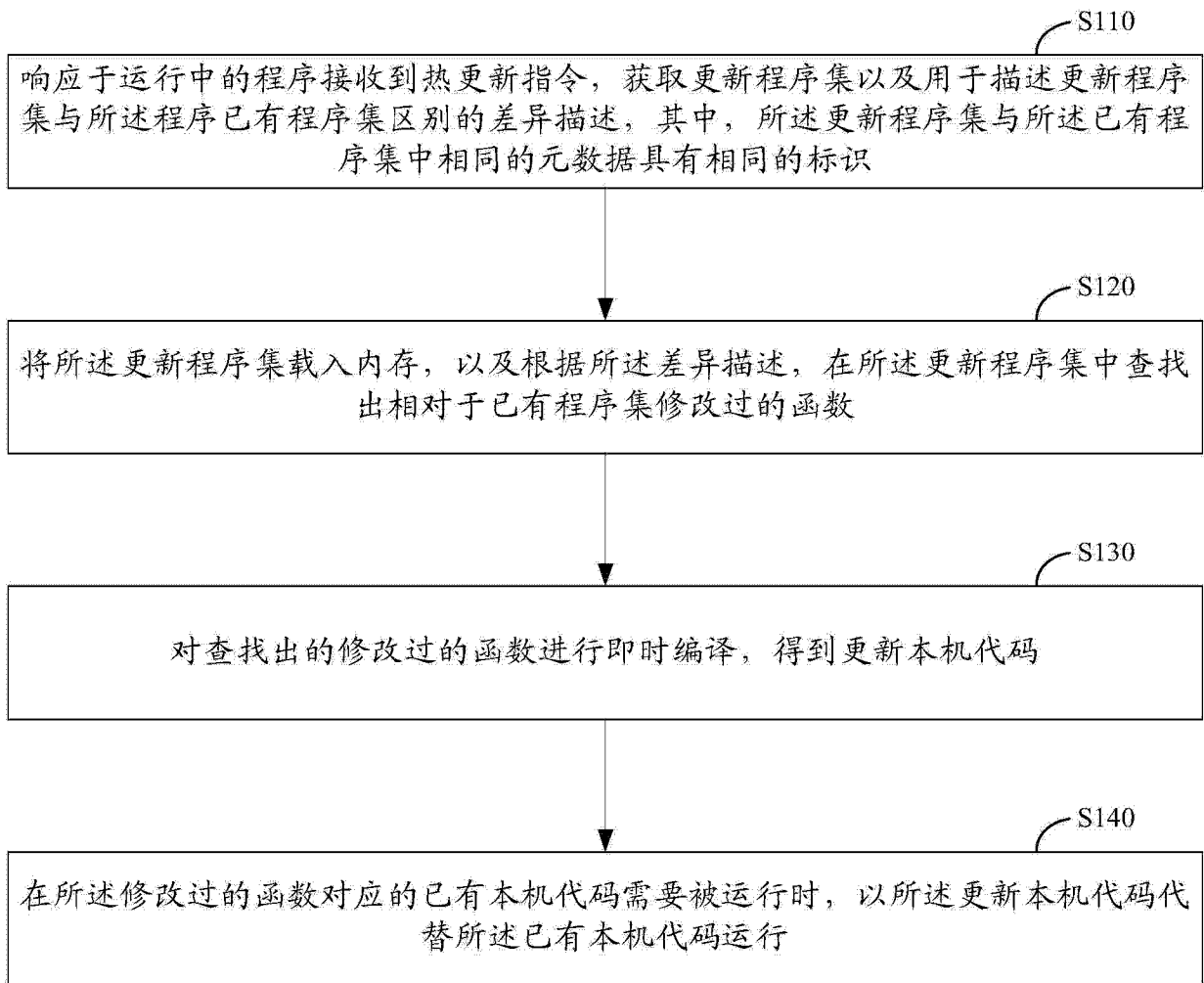


图 1

```
0A 00 00 00 EE 11 00 00 00 43 6C 61 73 73 4C 69  
62 72 61 72 79 31 2E 64 6C 6C 83 02 00 00 06 83  
03 00 00 06 83 06 00 00 06 83 07 00 00 06 83 09  
00 00 06 83 12 00 00 06 85 08 00 00 04 06 00 85  
09 00 00 04 06 00 85 0A 00 00 04 01 00
```

图 2

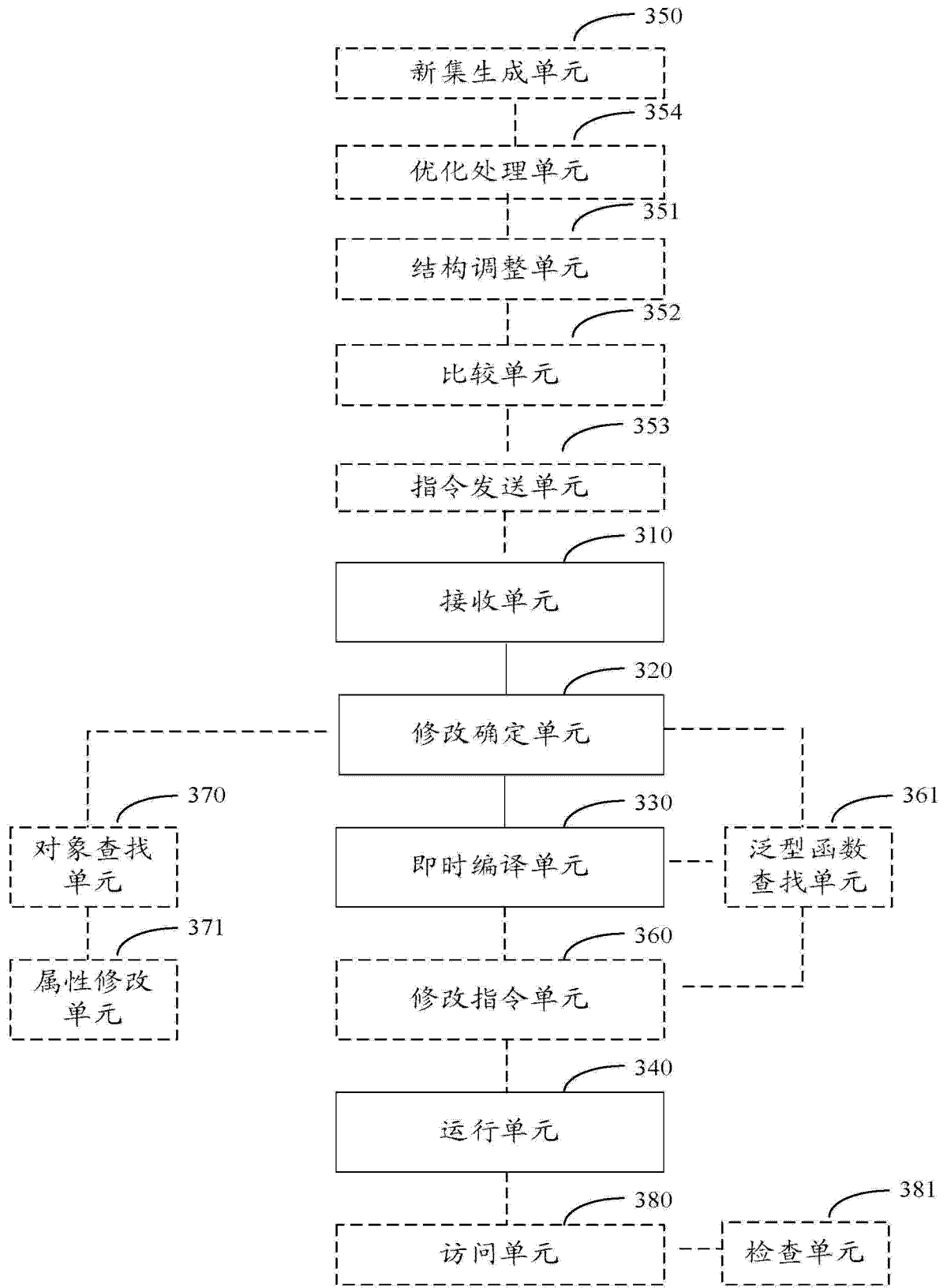


图 3