



(12) 发明专利

(10) 授权公告号 CN 113326041 B

(45) 授权公告日 2021. 11. 02

(21) 申请号 202110878229.5

CN 103093142 A, 2013.05.08

(22) 申请日 2021.08.02

CN 103914326 A, 2014.07.09

(65) 同一申请的已公布的文献号

CN 105183571 A, 2015.12.23

申请公布号 CN 113326041 A

US 2013014092 A1, 2013.01.10

(43) 申请公布日 2021.08.31

审查员 黄睿

(73) 专利权人 飞天诚信科技股份有限公司

地址 100085 北京市海淀区学清路9号汇智大厦B楼17层

(72) 发明人 陆舟 于华章

(51) Int. Cl.

G06F 8/30 (2018.01)

G06F 9/455 (2006.01)

(56) 对比文件

CN 103559073 A, 2014.02.05

CN 111273968 A, 2020.06.12

权利要求书4页 说明书19页 附图4页

(54) 发明名称

一种虚方法的调用方法、电子设备及计算机可读存储介质

(57) 摘要

本发明提供了一种虚方法的调用方法、电子设备及计算机可读存储介质,属于Java卡领域。Java卡根据虚方法调用指令确定拟被调用虚方法当前所属的类的地址,根据从虚拟机运行栈中的对象引用得到对象实际类的地址,当拟被调用虚方法当前所属的类的地址与对象实际类的地址不同时,在拟被调用虚方法当前所属的类的各级子类的虚方法映射表中查找拟被调用虚方法的索引,如查找不到,在拟被调用虚方法当前所属的类的父类中查找拟被调用虚方法的索引,查找到后根据拟被调用虚方法的索引从拟被调用虚方法当前所属的类的虚方法表中获取实际被调用虚方法的地址。本发明提高了Java卡的API的扩展性。



1. 一种虚方法的调用方法,适用于Java卡,其特征在于,包括如下步骤:

步骤101:Java卡根据虚方法调用指令在常量池中获取拟被调用虚方法,从所述拟被调用虚方法中获取拟被调用虚方法当前所属的类的引用和拟被调用虚方法的索引,根据所述拟被调用虚方法当前所属的类的引用确定拟被调用虚方法当前所属的类的地址;

步骤102:所述Java卡从虚拟机运行栈中获取对象引用,以所述对象引用为索引在保存的对象表中获取对象数据,从所述对象数据中获取对象实际类所属的包的标识和对对象实际类的引用,根据所述对象实际类所属的包的标识和所述对象实际类的引用得到对象实际类的地址;

步骤103:所述Java卡判断拟被调用虚方法当前所属的类的地址与对象实际类的地址是否相同,如果是,执行步骤109,否则,根据拟被调用虚方法当前所属的类的地址和对对象实际类的地址建立链表,执行步骤104;

步骤104:所述Java卡判断所述链表中是否存在拟被调用虚方法当前所属的类的直接子类的地址,如果是,执行步骤105,否则,执行步骤107;

步骤105:所述Java卡判断拟被调用虚方法当前所属的类的直接子类的虚方法映射表中是否存在拟被调用虚方法的索引,如果是,执行步骤106,否则,执行步骤107;

步骤106:所述Java卡将拟被调用虚方法当前所属的类的地址设置为所述拟被调用虚方法当前所属的类的直接子类的地址,将拟被调用虚方法的索引设置为拟被调用虚方法在拟被调用虚方法当前所属的类的直接子类的虚方法映射表中的索引,返回步骤104;

步骤107:所述Java卡判断拟被调用虚方法当前所属的类的虚方法表中是否存在所述拟被调用虚方法的索引,如果是,执行步骤109,否则,执行步骤108;

步骤108:所述Java卡从拟被调用虚方法当前所属的类中获取拟被调用虚方法当前所属的类的父类的引用,根据拟被调用虚方法当前所属的类的父类的引用确定所述拟被调用虚方法当前所属的类的地址,根据所述拟被调用虚方法当前所属的类的地址获取所述拟被调用虚方法当前所属的类,返回步骤107;

步骤109:所述Java卡根据所述拟被调用虚方法的索引从所述拟被调用虚方法当前所属的类的虚方法表中获取实际被调用虚方法的地址;

所述虚方法映射表中包括所述虚方法映射表所属的类从所述虚方法映射表所属的类的父类继承的虚方法的索引,还包括所述虚方法映射表所属的类包含的虚方法的索引。

2. 如权利要求1所述的方法,其特征在于,所述Java卡根据虚方法调用指令在常量池中获取拟被调用虚方法,具体为:所述Java卡从所述虚方法调用指令中获取指令操作数,以所述指令操作数为索引在常量池中获取拟被调用虚方法。

3. 如权利要求2所述的方法,其特征在于,所述以所述指令操作数为索引在常量池中获取拟被调用虚方法,具体包括:

步骤A1:所述Java卡从内存的全局变量中获取包信息地址列表的地址和拟被调用虚方法所属的包的标识,根据所述包信息地址列表的地址和所述拟被调用虚方法所属的包的标识获取所述拟被调用虚方法所属的包的常量池组件信息;

步骤A2:所述Java卡解析所述拟被调用虚方法所属的包的常量池组件信息得到常量池地址,获取所述常量池地址对应的数据得到常量池组件,以所述指令操作数为索引在所述常量池组件中获取所述拟被调用虚方法。

4. 如权利要求1所述的方法,其特征在于,所述从所述拟被调用虚方法中获取拟被调用虚方法当前所属的类的引用和拟被调用虚方法的索引,根据所述拟被调用虚方法当前所属的类的引用确定拟被调用虚方法当前所属的类的地址,具体包括:

步骤C1:所述Java卡从拟被调用虚方法中获取最后一个字节上的数据得到拟被调用虚方法的索引,从所述拟被调用虚方法当前所属的类中获取拟被调用虚方法当前所属的类的引用,判断所述拟被调用虚方法当前所属的类的引用的最高位是否为第二预设值,如果是,执行步骤C2,否则,执行步骤C3;

步骤C2:所述Java卡从所述拟被调用虚方法当前所属的类中获取拟被调用虚方法当前所属的类的父类的引用,从所述拟被调用虚方法当前所属的类的父类的引用中分别获取拟被调用虚方法当前所属的类的索引和拟被调用虚方法所属的包的索引,根据所述拟被调用虚方法所属的包的索引获取对应的导出表,根据所述拟被调用虚方法当前所属的类的索引在所述导出表中获取所述拟被调用虚方法当前所属的类的地址;

步骤C3:所述Java卡从拟被调用虚方法所属的包信息中获取拟被调用虚方法所属的包的类组件的基地址,根据所述拟被调用虚方法所属的包的类组件的基地址和所述拟被调用虚方法当前所属的类的地址偏移得到所述拟被调用虚方法当前所属的类的地址。

5. 如权利要求1所述的方法,其特征在于,所述以所述对象引用为索引在保存的对象表中获取对象数据,具体包括:

步骤B1:所述Java卡获取对象数据的基地址、对象数据的起始编号和对象数据地址的最大长度,将所述对象引用与所述对象数据的起始编号做减法运算得到第二运算值,将所述第二运算值与所述对象数据地址的最大长度做乘法运算得到第三运算值,将所述第三运算值与所述对象数据的基地址做加法运算得到所述对象数据的地址;

步骤B2:所述Java卡获取所述对象数据的地址对应的数据得到对象数据。

6. 如权利要求1所述的方法,其特征在于,所述根据所述对象实际类所属的包的标识和所述对象实际类的引用得到对象实际类的地址,具体包括:

步骤D1:所述Java卡以所述对象实际类所属的包的标识为索引在包信息地址列表中获取对象实际类所属的包地址信息,获取所述对象实际类所属的包地址信息对应的数据得到对象实际类所属的包信息,从所述对象实际类所属的包信息中获取前四个字节上的数据得到对象实际类所属的包的类组件的起始地址;

步骤D2:所述Java卡将所述对象实际类所属的包的类组件的起始地址和所述对象实际类的引用进行加法运算得到对象实际类的地址。

7. 如权利要求1所述的方法,其特征在于,所述根据拟被调用虚方法当前所属的类的地址和对象实际类的地址建立链表,具体包括:

步骤M1:所述Java卡建立链表;

步骤M2:所述Java卡将所述对象实际类的地址添加到所述链表中;

步骤M3:所述Java卡判断所述拟被调用虚方法当前所属的类的地址和所述对象实际类的地址是否相同,如果是,链表建立完成,否则,执行步骤M4;

步骤M4:所述Java卡将所述对象实际类的地址设置为所述对象实际类的父类的地址,返回步骤M2。

8. 如权利要求7所述的方法,其特征在于,所述Java卡判断所述链表中是否存在拟被

调用虚方法当前所属的类的直接子类的地址,具体为:所述Java卡根据所述拟被调用虚方法当前所属的类的地址在所述链表中查找拟所述拟被调用虚方法当前所属的类所在的节点,判断所述拟被调用虚方法当前所属的类所在的节点之后是否还存在节点,如果是,判定所述拟被调用虚方法当前所属的类存在直接子类,否则,判定所述拟被调用虚方法当前所属的类不存在直接子类。

9.如权利要求1所述的方法,其特征在于,所述步骤105之前还包括:所述Java卡从所述拟被调用虚方法当前所属的类中获取所述拟被调用虚方法所属的类的虚方法的个数和所述拟被调用虚方法所属的类的虚方法的起始编号并保存。

10.如权利要求9所述的方法,其特征在于,所述Java卡判断拟被调用虚方法当前所属的类的虚方法表中是否存在所述拟被调用虚方法的索引,具体为:所述Java卡根据所述拟被调用虚方法所属的类的虚方法的个数和所述拟被调用虚方法所属的类的虚方法的起始编号得到第四运算值,判断拟被调用虚方法的索引是否大于或者等于所述拟被调用虚方法所属的类的虚方法的起始编号且小于第四运算值,如果是,判定所述拟被调用虚方法当前所属的类的虚方法表中存在所述拟被调用虚方法的索引,否则,判定所述拟被调用虚方法当前所属的类的虚方法表中不存在所述拟被调用虚方法的索引。

11.如权利要求9所述的方法,其特征在于,所述步骤109具体为:所述Java卡根据拟被调用虚方法的索引和所述拟被调用虚方法所属的类的虚方法的起始编号进行减法运算得到第五运算值,以所述第五运算值为索引在所述拟被调用虚方法当前所属的类的虚方法表中获取实际被调用虚方法的地址。

12.如权利要求1所述的方法,其特征在于,所述步骤108具体包括:

步骤N1:所述Java卡从拟被调用虚方法当前所属的类中获取拟被调用虚方法当前所属的类的父类的引用,判断所述拟被调用虚方法当前所属的类的父类的引用的最高位是否为第二预设值,如果是,执行步骤N2,否则,执行步骤N3;

步骤N2:所述Java卡从拟被调用虚方法当前所属的类的父类的引用中获取拟被调用虚方法当前所属的类的父类的索引和拟被调用虚方法当前所属的类的父类所属的包的索引并分别作为拟被调用虚方法当前所属的类的索引和拟被调用虚方法所属的包的索引,根据所述拟被调用虚方法所属的包的索引获取对应的导出表,根据所述拟被调用虚方法当前所属的类的索引在所述导出表中获取拟被调用虚方法当前所属的类的地址,返回步骤107;

步骤N3:所述Java卡从被调用虚方法所属的包信息中获取拟被调用虚方法所属的包的类组件的基地址,根据拟被调用虚方法所属的包的类组件基地址和拟被调用虚方法当前所属的类的地址偏移得到拟被调用虚方法当前所属的类的地址,返回步骤107。

13.如权利要求1所述的方法,其特征在于,还包括所述虚方法映射表的生成过程,包括如下步骤:

步骤S1:转换工具从类组件中获取一个类记为当前类,从当前类中获取当前类的虚方法的起始编号和当前类的虚方法的个数;

步骤S2:所述转换工具根据所述当前类的虚方法的起始编号和所述当前类的虚方法的个数创建当前类的虚方法映射表,对当前类的虚方法映射表进行初始化;

步骤S3:所述转换工具从当前类中获取当前类的父类的引用,判断当前类的父类的引用的最高位是否为第二预设值,如果是,执行步骤S4,否则,执行步骤S5;

步骤S4:所述转换工具从当前类的父类的引用中分别获取当前类的父类的索引和当前类的父类所属的包的索引,获取预存的导出表地址,根据当前类的父类所属的包的索引和所述导出表地址获取当前类的父类,执行步骤S6;

步骤S5:所述转换工具获取预存的当前类所属的包的类组件的基地址,根据当前类所属的包的类组件的基地址和当前类的父类的引用得到当前类的父类,执行步骤S6;

步骤S6:所述转换工具从当前类的父类中获取当前类的父类的虚方法的起始编号和当前类的父类的虚方法的个数,根据当前类的父类的虚方法的起始编号和当前类的父类的虚方法的个数更新当前类的虚方法映射表;

步骤S7:所述转换工具将当前类的虚方法映射表添加到当前类中得到包含虚方法映射表的当前类。

14. 一种电子设备,其特征在于,包括:处理器和存储器;

所述存储器,用于存储计算机程序;

所述处理器,用于运行所述存储器存储的计算机程序来执行权利要求1-13任一项所述的方法。

15. 一种计算机可读存储介质,其特征在于,用于存储计算机程序,当所述计算机程序在计算机上运行时,使得所述计算机执行如权利要求1-13任一项所述的方法。

一种虚方法的调用方法、电子设备及计算机可读存储介质

技术领域

[0001] 本发明涉及Java卡领域,尤其涉及一种虚方法的调用方法、电子设备及计算机可读存储介质。

背景技术

[0002] Java卡是一种可以运行Java程序的智能卡,其通过解释执行应用文件包中Java字节码来实现应用的功能。Java字节码的生成过程包括:使用Java卡的API编写Java应用代码,编译Java应用代码得到class字节码,通过转换工具将class字节码转换为可以被Java卡执行的Java字节码,Java卡通过虚方法调用指令来执行Java字节码。

[0003] 现有技术中,Java卡通过虚方法调用指令查找被调用虚方法时,只能通过子类索引父类的方式进行,这种方式的缺点在于一旦父类所属的应用文件包升级,其子类所属的应用文件包必须重新生成Java字节码才能继续运行,这是因为在父类所属的应用文件包升级后,父类所属的应用文件中的虚方法的地址将会发生改变,这就Java卡的API的扩展带来了极大的不便。

发明内容

[0004] 为了解决现有技术中的不足,本发明提供了一种虚方法的调用方法、电子设备及计算机可读存储介质。

[0005] 第一方面,本发明提供了一种虚方法的调用方法,包括如下步骤:

[0006] 步骤101:Java卡根据虚方法调用指令在常量池中获取拟被调用虚方法,从所述拟被调用虚方法中获取拟被调用虚方法当前所属的类的引用和拟被调用虚方法的索引,根据所述拟被调用虚方法当前所属的类的引用确定拟被调用虚方法当前所属的类的地址;

[0007] 步骤102:所述Java卡从虚拟机运行栈中获取对象引用,以所述对象引用为索引在保存的对象表中获取对象数据,从所述对象数据中获取对象实际类所属的包的标识和对象实际类的引用,根据所述对象实际类所属的包的标识和所述对象实际类的引用得到对象实际类的地址;

[0008] 步骤103:所述Java卡判断拟被调用虚方法当前所属的类的地址与对象实际类的地址是否相同,如果是,执行步骤109,否则,根据拟被调用虚方法当前所属的类的地址和对象实际类的地址建立链表,执行步骤104;

[0009] 步骤104:所述Java卡判断所述链表中是否存在拟被调用虚方法当前所属的类的直接子类的地址,如果是,执行步骤105,否则,执行步骤107;

[0010] 步骤105:所述Java卡判断拟被调用虚方法当前所属的类的直接子类的虚方法映射表中是否存在所述拟被调用虚方法的索引,如果是,执行步骤106,否则,执行步骤107;

[0011] 步骤106:所述Java卡将拟被调用虚方法当前所属的类的地址设置为所述拟被调用虚方法当前所属的类的直接子类的地址,将拟被调用虚方法的索引设置为拟被调用虚方法在拟被调用虚方法当前所属的类的直接子类的虚方法映射表中的索引,返回步骤104;

[0012] 步骤107:所述Java卡判断拟被调用虚方法当前所属的类的虚方法表中是否存在所述拟被调用虚方法的索引,如果是,执行步骤109,否则,执行步骤108;

[0013] 步骤108:所述Java卡从拟被调用虚方法当前所属的类中获取拟被调用虚方法当前所属的类的父类的引用,根据拟被调用虚方法当前所属的类的父类的引用确定所述拟被调用虚方法当前所属的类的地址,根据所述拟被调用虚方法当前所属的类的地址获取所述拟被调用虚方法当前所属的类,返回步骤107;

[0014] 步骤109:所述Java卡根据所述拟被调用虚方法的索引从所述拟被调用虚方法当前所属的类的虚方法表中获取实际被调用虚方法的地址;

[0015] 所述虚方法映射表包括从所述虚方法映射表所属的类的父类继承的虚方法的索引和所述虚方法映射表所属的类的包含的虚方法的索引。

[0016] 第二方面,本发明提供了一种电子设备,包括:存储器和处理器;

[0017] 存储器,用于存储计算机程序;

[0018] 处理器,用于运行所述存储器存储的计算机程序来执行本申请中的虚方法的调用方法。

[0019] 第三方面,本发明提供了一种计算机可读存储介质,用于存储计算机程序,当计算机程序在计算机上运行时,使得所述计算机执行本申请中的虚方法的调用方法。

[0020] 本发明提供了一种虚方法的调用方法、电子设备及计算机可读存储介质,在父类所属的应用文件包升级后,其子类所属的应用文件包无需重新生成Java字节码仍可正常运行,提高了Java卡的API的扩展性。

附图说明

[0021] 图1为本发明实施例1提供了一种虚方法的调用方法流程图;

[0022] 图2为本发明实施例2提供了一种生成虚方法映射表的方法流程图;

[0023] 图3为本发明实施例3提供了一种虚方法的调用方法流程图;

[0024] 图4为本发明具体实施方式提供了一种电子设备的模块图。

具体实施方式

[0025] 本申请提出了一种虚方法的调用方法、电子设备及计算机可读存储介质,下面结合附图,对本申请具体实施方式进行详细说明。所述实施例的示例在附图中示出。下面通过参考附图描述的实施例是示例性的,仅用于解释本申请,而不能解释为对本申请的限制。

[0026] 本技术领域技术人员可以理解,除非另外定义,这里使用的所有术语(包括技术术语和科学术语),具有与本申请所属领域中的普通技术人员的一般理解相同的意义。还应该理解的是,诸如通用字典中定义的那些术语,应该被理解为具有与现有技术的上下文中的意义一致的意义,并且除非像这里一样被特定定义,否则不会用理想化或过于正式的含义来解释。

[0027] 为使本发明的目的、技术方案和优点更加清楚,下面结合附图对本发明实施方式作进一步地详细描述。

[0028] 实施例1

[0029] 本发明实施例1提供了一种虚方法的调用方法,如图1所示,该方法包括如下步骤:

[0030] 步骤101:Java卡根据虚方法调用指令在常量池中获取拟被调用虚方法,从拟被调用虚方法中获取拟被调用虚方法当前所属的类的引用和拟被调用虚方法的索引,根据拟被调用虚方法当前所属的类的引用确定拟被调用虚方法当前所属的类的地址;

[0031] 步骤102:Java卡从虚拟机运行栈中获取对象引用,以对象引用为索引在保存的对象表中获取对象数据,从对象数据中获取对象实际类所属的包的标识和对象实际类的引用,根据对象实际类所属的包的标识和对象实际类的引用得到对象实际类的地址;

[0032] 步骤103:Java卡判断拟被调用虚方法当前所属的类的地址与对象实际类的地址是否相同,如果是,执行步骤109,否则,根据拟被调用虚方法当前所属的类的地址和对象实际类的地址建立链表,执行步骤104;

[0033] 步骤104:Java卡判断链表中是否存在拟被调用虚方法当前所属的类的直接子类的地址,如果是,执行步骤105,否则,执行步骤107;

[0034] 步骤105:Java卡判断拟被调用虚方法当前所属的类的直接子类的虚方法映射表中是否存在拟被调用虚方法的索引,如果是,执行步骤106,否则,执行步骤107;

[0035] 步骤106:Java卡将拟被调用虚方法当前所属的类的地址设置为拟被调用虚方法当前所属的类的直接子类的地址,将拟被调用虚方法的索引设置为拟被调用虚方法在拟被调用虚方法当前所属的类的直接子类的虚方法映射表中的索引,返回步骤104;

[0036] 步骤107:Java卡判断拟被调用虚方法当前所属的类的虚方法表中是否存在拟被调用虚方法的索引,如果是,执行步骤109,否则,执行步骤108;

[0037] 步骤108:Java卡从拟被调用虚方法当前所属的类中获取拟被调用虚方法当前所属的类的父类的引用,根据拟被调用虚方法当前所属的类的父类的引用确定拟被调用虚方法当前所属的类的地址,根据拟被调用虚方法当前所属的类的地址获取拟被调用虚方法当前所属的类,返回步骤107;

[0038] 步骤109:Java卡根据拟被调用虚方法的索引从拟被调用虚方法当前所属的类的虚方法表中获取实际被调用虚方法的地址;

[0039] 虚方法映射表包括从虚方法映射表所属的类的父类继承的虚方法的索引和虚方法映射表所属的类包含的虚方法的索引。

[0040] 可选地,本实施例中,Java卡根据虚方法调用指令在常量池中获取拟被调用虚方法,具体为:Java卡从虚方法调用指令中获取指令操作数,以指令操作数为索引在常量池中获取拟被调用虚方法。

[0041] 可选地,本实施例中,以指令操作数为索引在常量池中获取拟被调用虚方法,具体包括:

[0042] 步骤A1:Java卡从内存的全局变量中获取包信息地址列表的地址和拟被调用虚方法所属的包的标识,根据包信息地址列表的地址和拟被调用虚方法所属的包的标识获取拟被调用虚方法所属的包的常量池组件信息;

[0043] 步骤A2:Java卡解析拟被调用虚方法所属的包的常量池组件信息得到常量池地址,获取常量池地址对应的数据得到常量池组件,以指令操作数为索引在常量池组件中获取拟被调用虚方法。

[0044] 可选地,本实施例中,从拟被调用虚方法中获取拟被调用虚方法当前所属的类的引用和拟被调用虚方法的索引,根据拟被调用虚方法当前所属的类的引用确定拟被调用虚

方法当前所属的类的地址,具体包括:

[0045] 步骤C1:Java卡从拟被调用虚方法中获取最后一个字节上的数据得到拟被调用虚方法的索引,从拟被调用虚方法当前所属的类中获取拟被调用虚方法当前所属的类的引用,判断拟被调用虚方法当前所属的类的引用的最高位是否为第二预设值,如果是,执行步骤C2,否则,执行步骤C3;

[0046] 步骤C2:Java卡从拟被调用虚方法当前所属的类中获取拟被调用虚方法当前所属的类的父类的引用,从拟被调用虚方法当前所属的类的父类的引用中分别获取拟被调用虚方法当前所属的类的索引和拟被调用虚方法所属的包的索引,根据拟被调用虚方法所属的包的索引获取对应的导出表,根据拟被调用虚方法当前所属的类的索引在导出表中获取拟被调用虚方法当前所属的类的地址;

[0047] 步骤C3:Java卡从拟被调用虚方法所属的包信息中获取拟被调用虚方法所属的包的类组件的基地址,根据拟被调用虚方法所属的包的类组件的基地址和拟被调用虚方法当前所属的类的地址偏移得到拟被调用虚方法当前所属的类的地址。

[0048] 可选地,本实施例中,以对象引用为索引在保存的对象表中获取对象数据,具体包括:

[0049] 步骤B1:Java卡获取对象数据的基地址、对象数据的起始编号和对象数据地址的最大长度,将对象引用与对象数据的起始编号做减法运算得到第二运算值,将第二运算值与对象数据地址的最大长度做乘法运算得到第三运算值,将第三运算值与对象数据的基地址做加法运算得到对象数据的地址;

[0050] 步骤B2:Java卡获取对象数据的地址对应的数据得到对象数据。

[0051] 可选地,本实施例中,根据对象实际类所属的包的标识和对象实际类的引用得到对象实际类的地址,具体包括:

[0052] 步骤D1:Java卡以对象实际类所属的包的标识为索引在包信息地址列表中获取对象实际类所属的包地址信息,获取对象实际类所属的包地址信息对应的数据得到对象实际类所属的包信息,从对象实际类所属的包信息中获取前四个字节上的数据得到对象实际类所属的包的类组件的起始地址;

[0053] 步骤D2:Java卡将对象实际类所属的包的类组件的起始地址和对象实际类的引用进行加法运算得到对象实际类的地址。

[0054] 可选地,本实施例中,根据拟被调用虚方法当前所属的类的地址和对象实际类的地址建立链表,具体包括:

[0055] 步骤M1:Java卡建立链表;

[0056] 步骤M2:Java卡将对象实际类的地址添加到链表中;

[0057] 步骤M3:Java卡判断拟被调用虚方法当前所属的类的地址和对象实际类的地址是否相同,如果是,链表建立完成,否则,执行步骤M4;

[0058] 步骤M4:Java卡将对象实际类的地址设置为对象实际类的父类的地址,返回步骤M2。

[0059] 可选地,本实施例中,Java卡判断链表中是否存在拟被调用虚方法当前所属的类的直接子类的地址,具体为:Java卡根据拟被调用虚方法当前所属的类的地址在链表中查找拟被调用虚方法当前所属的类所在的节点,判断拟被调用虚方法当前所属的类所在的节

点之后是否还存在节点,如果是,判定拟被调用虚方法当前所属的类存在直接子类,否则,判定拟被调用虚方法当前所属的类不存在直接子类。

[0060] 可选地,本实施例中,步骤105之前还包括:Java卡从拟被调用虚方法当前所属的类中获取拟被调用虚方法所属的类的虚方法的个数和拟被调用虚方法所属的类的虚方法的起始编号并保存。

[0061] 可选地,本实施例中,Java卡判断拟被调用虚方法当前所属的类的虚方法表中是否存在拟被调用虚方法的索引,具体为:Java卡根据拟被调用虚方法所属的类的虚方法的个数和拟被调用虚方法所属的类的虚方法的起始编号得到第四运算值,判断拟被调用虚方法的索引是否大于或者等于拟被调用虚方法所属的类的虚方法的起始编号且小于第四运算值,如果是,判定拟被调用虚方法当前所属的类的虚方法表中存在拟被调用虚方法的索引,否则,判定拟被调用虚方法当前所属的类的虚方法表中不存在拟被调用虚方法的索引。

[0062] 可选地,本实施例中,步骤108具体包括:

[0063] 步骤N1:Java卡从拟被调用虚方法当前所属的类中获取拟被调用虚方法当前所属的类的父类的引用,判断拟被调用虚方法当前所属的类的父类的引用的最高位是否为第二预设值,如果是,执行步骤N2,否则,执行步骤N3;

[0064] 步骤N2:Java卡从拟被调用虚方法当前所属的类的父类的引用中获取拟被调用虚方法当前所属的类的父类的索引和拟被调用虚方法当前所属的类的父类所属的包的索引并分别作为拟被调用虚方法当前所属的类的索引和拟被调用虚方法所属的包的索引,根据拟被调用虚方法所属的包的索引获取对应的导出表,根据拟被调用虚方法当前所属的类的索引在导出表中获取拟被调用虚方法当前所属的类的地址,返回步骤107;

[0065] 步骤N3:Java卡从被调用虚方法所属的包信息中获取拟被调用虚方法所属的包的类组件的基地址,根据拟被调用虚方法所属的包的类组件基地址和拟被调用虚方法当前所属的类的地址偏移得到拟被调用虚方法当前所属的类的地址,返回步骤107。

[0066] 可选地,本实施例中,步骤109具体为:Java卡根据拟被调用虚方法的索引和拟被调用虚方法所属的类的虚方法的起始编号进行减法运算得到第五运算值,以第五运算值为索引在拟被调用虚方法当前所属的类的虚方法表中获取实际被调用虚方法。

[0067] 可选地,本实施例中,还包括所述虚方法映射表的生成过程,包括如下步骤:

[0068] 步骤S1:转换工具从类组件中获取一个类记为当前类,从当前类中获取当前类的虚方法的起始编号和当前类的虚方法的个数;

[0069] 步骤S2:转换工具根据所述当前类的虚方法的起始编号和所述当前类的虚方法的个数创建当前类的虚方法映射表,对当前类的虚方法映射表进行初始化;

[0070] 步骤S3:转换工具从当前类中获取当前类的父类的引用,判断当前类的父类的引用的最高位是否为第二预设值,如果是,执行步骤S4,否则,执行步骤S5;

[0071] 步骤S4:转换工具从当前类的父类的引用中分别获取当前类的父类的索引和当前类的父类所属的包的索引,获取预存的导出表地址,根据当前类的父类所属的包的索引和所述导出表地址获取当前类的父类,执行步骤S6;

[0072] 步骤S5:转换工具获取预存的当前类所属的包的类组件的基地址,根据当前类所属的包的类组件的基地址和当前类的父类的引用得到当前类的父类,执行步骤S6;

[0073] 步骤S6:转换工具从当前类的父类中获取当前类的父类的虚方法的起始编号和当

前类的父类的虚方法的个数,根据当前类的父类的虚方法的起始编号和当前类的父类的虚方法的个数更新当前类的虚方法映射表;

[0074] 步骤S7:转换工具将当前类的虚方法映射表添加到当前类中得到包含虚方法映射表的当前类。

[0075] 本实施例提供了一种虚方法的调用方法,通过虚方法映射表来索引原应用和新应用中的虚方法,在父类所属的应用文件包升级后,可以使子类所属的应用文件包无需重新生成Java字节码仍可正常运行,使得虚方法映射表既可以服务原应用又可以服务新应用,提高了Java卡的API的扩展性。

[0076] 实施例2

[0077] 本发明实施例2提供了一种生成虚方法映射表的方法,如图2所示,包括如下步骤:

[0078] 步骤201:转换工具根据类组件组成结构从类组件中获取类组件包含的全部类并保存在缓存区;

[0079] 具体地,每个类组件是通过一个类组件组成结构来表示的;

[0080] 类组件组成结构具体包括:类组件标签、类组件大小、签名池长度、签名池、接口列表和类列表,其中,类列表中存储有该包中包含的全部类;

[0081] 可选地,本实施例中,步骤201具体为:转换工具解析类组件得到类组件标签、类组件大小、签名池长度、签名池、接口列表和类列表,将类列表保存到缓存区。

[0082] 步骤202:转换工具从缓存区中获取一个类记为当前类,从当前类中获取当前类的虚方法的起始编号和当前类的虚方法的个数;

[0083] 可选地,类组件中的每个类是通过一个类信息结构来表示的;

[0084] 一个类信息结构具体包括:位阈、超类结构、实例域大小、起始引用类型的实例域的索引值、引用类型的实例域的个数、类的虚方法的起始编号、类的虚方法的个数、包的虚方法的起始编号、包的虚方法的个数、包的公共属性或覆盖属性的虚方法映射表、包的可视虚方法映射表、被实现的接口列表和被实现的远程接口;

[0085] 转换工具从缓存区中获取一个类记为当前类,具体为:转换工具在类列表中截取一个类信息结构,将截取到的类信息结构记为当前类;

[0086] 例如:当前类具体为:06 00 00 01 00 00 80 00 00 FF 00

[0087] 01 /*base*/

[0088] 04 /*count*/

[0089] 00 08

[0090] 00 0C

[0091] 00 10

[0092] 00 15

[0093] 当前类的虚方法的起始编号为01,当前类的虚方法的个数为04;

[0094] 步骤203:转换工具根据当前类的虚方法的起始编号和当前类的虚方法的个数创建当前类的虚方法映射表,对当前类的虚方法映射表进行初始化;

[0095] 可选地,本实施例中,步骤203具体为:转换工具对当前类的虚方法的起始编号和当前类的虚方法的个数进行求和运算得到第一长度值,创建长度为第一长度值的虚方法映射表作为当前类的虚方法映射表,将当前类的虚方法映射表中的各项初始化为第一预设

值;

[0096] 可选地,本实施例中,第一预设值具体为:FF;

[0097] 例如:当前类的虚方法的起始编号为01,当前类的虚方法的个数为04,第一长度值为05,第一预设值为FF;创建的当前类的虚方法映射表为:00 00 00 00 00,对当前类的虚方法映射表进行初始化得到的虚方法映射表为:FF FF FF FF FF;

[0098] 步骤204:转换工具从当前类中获取当前类的父类的引用,判断当前类的父类的引用的最高位是否为第二预设值,如果是,执行步骤205,否则,执行步骤207;

[0099] 可选地,本实施例中,转换工具从当前类中获取当前类的父类的引用,具体为:转换工具从当前类的类信息结构中获取第二个字节和第三个字节上的数据得到当前类的父类的引用;

[0100] 可选地,本实施例中,第二预设值具体为1;

[0101] 例如:当前类的父类的引用具体为:00 00,其最高位不是第二预设值,执行步骤207;

[0102] 步骤205:转换工具从当前类的父类的引用中分别获取当前类的父类的索引和当前类的父类所属的包的索引;

[0103] 可选地,本实施例中,步骤205具体为:转换工具从当前类的父类的引用中获取第0位至第7位得到当前类的父类的索引,获取第8位至第14位得到当前类的父类所属的包的索引;

[0104] 步骤206:转换工具获取预存的导出表地址列表,根据当前类的父类所属的包的索引从导出表地址列表中获取当前类的父类所属的包的索引对应的导出表地址,根据当前类的父类所属的包的索引对应的导出表地址获取对应的导出表,根据当前类的父类的索引在导出表中查找当前类的父类的地址偏移,根据当前类的父类所属的包的类组件基地址和当前类的父类的地址偏移得到当前类的父类的地址,根据当前类的父类的地址获取当前类的父类,执行步骤208;

[0105] 步骤207:转换工具获取预存的当前类所属的包的类组件的基地址,根据当前类所属的包的类组件的基地址和当前类的父类的引用得到当前类的父类的地址,根据当前类的父类的地址获取当前类的父类,执行步骤208;

[0106] 可选地,本实施例中,根据当前类所属的包的类组件的基地址和当前类的父类的引用得到当前类的父类的地址,具体为:转换工具将当前类所属的包的类组件的基地址和当前类的父类的引用进行加法运算得到当前类的父类的地址;

[0107] 例如:当前类所属的包的类组件的基地址具体为:00 00 01 00;

[0108] 当前类的父类的地址具体为:00 00 01 00;

[0109] 当前类的父类具体为:00 00 01 00;

[0110] 步骤208:转换工具从当前类的父类中获取当前类的父类的虚方法的起始编号和当前类的父类的虚方法的个数;

[0111] 例如:当前类的父类的虚方法的起始编号具体为00,当前类的父类的虚方法的个数具体为01;

[0112] 步骤209:转换工具根据当前类的父类的虚方法的起始编号和当前类的父类的虚方法的个数更新当前类的虚方法映射表;

[0113] 可选地,本实施例中,步骤209具体为:转换工具对前类的父类的虚方法的起始编号和当前类的父类的虚方法的个数进行求和运算,将求和运算得到的结果与第二预设值做减法运算得到第六运算值,将当前类的虚方法映射表的第0项至第六运算值项对应更新为0至第六运算值;

[0114] 可选地,本实施例中,第二预设值具体为:1;

[0115] 例如:第六运算值具体为:0;

[0116] 更新后的当前类的虚方法映射表具体为:00 FF FF FF FF;

[0117] 步骤210:转换工具将当前类的虚方法映射表添加到当前类中得到包含虚方法映射表的当前类,执行步骤211;

[0118] 可选地,本实施例中,将当前类的虚方法映射表添加到当前类中得到包含虚方法映射表的当前类,具体为:转换工具将当前类的虚方法映射表添加到当前类的最后一个虚方法之后得到包含虚方法映射表的当前类;

[0119] 可选地,本实施例中,虚方法映射表的第0项至第六运算值项的虚方法为虚方法映射表从其所属的类的父类中继承的虚方法,其余的虚方法为属于当前类的虚方法;

[0120] 例如:包含虚方法映射表的当前类具体为:

[0121] 06 00 00 01 00 00 80 00 00 FF 00

[0122] 01 /*base*/

[0123] 04 /*count*/

[0124] 00 08

[0125] 00 0C

[0126] 00 10

[0127] 00 15

[0128] 00 FF FF FF FF /*已经创建好的VMMT*/

[0129] 步骤211:转换工具判断缓存区中是否还存在未获取过的类,如果是,返回步骤202,否则,结束。

[0130] 本实施例还给出了步骤204中当转换工具判断当前类的父类的引用的最高位是第二预设值时,执行步骤205和步骤206的实例,具体如下:

[0131] 当前类的虚方法的起始编号为03,当前类的虚方法的个数为04,第一长度值为07,第一预设值为FF;创建的当前类的虚方法映射表为:00 00 00 00 00 00 00,对当前类的虚方法映射表进行初始化得到的虚方法映射表为:FF FF FF FF FF FF FF,当前类的父类的引用具体为:80 03,其最高位是第二预设值,执行步骤205;

[0132] 当前类的索引为03,当前类所属的包的索引为00;

[0133] 导出表地址数组具体为:

[0134] 00 00 1A 87

[0135] 00 00 1E 06

[0136] 00 00 20 32

[0137] 00 00 22 87

[0138] 00 00 43 02

[0139] 00 00 45 34

[0140] 当前类所属的包的索引对应的导出表地址具体为:00 00 1A 87;

[0141] 获取的导出表具体为:

[0142]

[0143] 0A

[0144] 01 07 //size

[0145] 0D //class_count

[0146] //第0个

[0147] 00 E2 //class_offset

[0148] 00 01 //static_field_count

[0149] 00 01 //static_method_count

[0150] //第1个

[0151] 01 D2 //class_offset

[0152] 00 02 //static_field_count

[0153] 00 03 //static_method_count

[0154] //第2个

[0155] 01 FE //class_offset

[0156] 00 0A //static_field_count

[0157] 00 03 //static_method_count

[0158] //第3个

[0159] 03 02 //class_offset

[0160] 00 03 //static_field_count

[0161] 00 08 //static_method_count

[0162]

[0163] 当前类的父类的索引为03,以03为索引在导出表中可以找到当前类的父类的地址偏移为03 02;

[0164] 当前类的父类所属的包的类组件基地址具体为:00 00 02 50,当前类的父类的地址具体为:00 00 05 52;

[0165] 当前类的父类具体为:06 80 03 02 00 00 80 00 00 FF 00

[0166] 03 /*base*/

[0167] 04 /*count*/

[0168] 00 10

[0169] 00 15

[0170] 00 1A

[0171] 00 2C

[0172] 当前类的父类的虚方法的起始编号具体为01,当前类的虚方法的个数具体为04;

[0173] 当前类的父类的虚方法映射表具体为:FF FF FF FF FF FF FF;

[0174] 更新后的当前类的父类的虚方法映射表具体为:00 01 02 03 04 FF FF;

[0175] 包含虚方法映射表的当前类的父类具体为:

[0176] 06 80 03 02 00 00 80 00 00 FF 00

[0177] 03 /*base*/
[0178] 04 /*count*/
[0179] 00 10
[0180] 00 15
[0181] 00 1A
[0182] 00 2C
[0183] 00 01 02 03 04 FF FF/*已经创建好的VMMT*/

[0184] 本实施例提供了一种生成虚方法映射表的方法,实现了为没有虚方法映射表的类创建虚方法映射表,使得在父类所属的应用文件包升级后,其子类所属的应用文件包无需重新生成Java字节码,Java卡可通过创建的虚方法映射表来查找被调用虚方法,提高了Java卡的API的扩展性。

[0185] 实施例3

[0186] 本发明实施例3提供了一种虚方法的调用方法,其中,本实施例中涉及的类均包含虚方法映射表,各虚方法映射表均是根据实施例2的生成虚方法映射表的方法生成的,本实施例的方法如图3所示,包括如下步骤:

[0187] 步骤301:Java卡从虚方法调用指令中获取指令操作数,以指令操作数为索引在常量池中获取拟被调用虚方法;

[0188] 可选地,本实施例中,虚方法调用指令的长度为3个字节,其中,第一个字节为指令操作码,第二个字节和第三个字节为指令操作数;

[0189] 相应地,本实施例中,Java卡从虚方法调用指令中获取指令操作数具体为:Java卡从虚方法调用指令中获取第二个字节和第三个字节上的数据得到指令操作数;

[0190] 例如:虚方法调用指令具体为:8B 00 2c;

[0191] 指令操作码具体为:8B;

[0192] 指令操作数具体为:002c;

[0193] 可选地,本实施例中,以指令操作数为索引在常量池中获取拟被调用虚方法,具体包括:

[0194] 步骤A1:Java卡从内存的全局变量中获取包信息地址列表的地址和拟被调用虚方法所属的包的标识,根据包信息地址列表的地址和拟被调用虚方法所属的包的标识获取拟被调用虚方法所属的包的常量池组件信息;

[0195] 可选地,根据包信息地址列表的地址和拟被调用虚方法所属的包的标识获取拟被调用虚方法所属的包的常量池组件信息,具体为:Java卡根据包信息地址列表的地址获取包信息地址列表,根据拟被调用虚方法所属的包的标识从包信息地址列表中获取拟被调用虚方法所属的包的地址,获取拟被调用虚方法所属的包的地址对应的数据得到拟被调用虚方法所属的包信息,根据包信息的数据结构解析拟被调用虚方法所属的包信息得到拟被调用虚方法所属的包的常量池组件信息;

[0196] 例如:拟被调用虚方法所属的包的标识具体为:05;

[0197] 包信息地址列表的地址具体为:000257be;

[0198] 包信息地址列表具体为:00 01 00 00 //包0的地址

[0199] 00 01 20 00 //包1的地址

- [0200] 00 01 30 00 //包2的地址
- [0201] 00 02 40 00 //包3的地址
- [0202] 00 02 50 00 //包4的地址
- [0203] 00 02 57 d2 //包5的地址
- [0204] ……;
- [0205] 拟被调用虚方法所属的包的地址具体为:00 02 57 d2;
- [0206] 拟被调用虚方法所属的包信息具体为:00 00 02 50 00 00 00 24 00 00 4c 00 00 00 00 98 00 00 64 5c 00 00 01 f3 00 00 88 00 00 00 01 ed 00 00 89 a4 00 00 01 18;
- [0207] 拟被调用虚方法所属的包的常量池组件信息具体为:00 00 89 a4 00 00 01 18;
- [0208] 步骤A2:Java卡解析拟被调用虚方法所属的包的常量池组件信息得到常量池地址,获取常量池地址对应的数据得到常量池组件,以指令操作数为索引在常量池组件中获取拟被调用虚方法;
- [0209] 可选地,本实施例中,Java卡解析拟被调用虚方法所属的包的常量池组件信息得到常量池地址,具体为:Java卡从拟被调用虚方法所属的包的常量池组件信息中获取前四个字节上的数据得到常量池地址;
- [0210] 例如:常量池地址具体为:00 00 89 a4;
- [0211] 常量池组件具体为:03 80 03 02 03 80 03 07 03 80 03 08 03 80 03 06 04 80 03 06 04 80 03 07 04 80 03 08 04 80 03 06 05 80 04 06 05 80 04 07 05 80 04 08 06 80 04 06 06 80 05 06 06 80 05 07 06 80 05 08 06 80 05 06;
- [0212] 被调用虚方法具体为:03 80 03 02;
- [0213] 步骤302:Java卡从拟被调用虚方法中获取拟被调用虚方法当前所属的类的引用和拟被调用虚方法的索引,根据拟被调用虚方法当前所属的类的引用确定拟被调用虚方法当前所属的类的地址;
- [0214] 可选地,本实施例中,步骤302具体包括:
- [0215] 步骤C1:Java卡从拟被调用虚方法中获取最后一个字节上的数据得到拟被调用虚方法的索引,从拟被调用虚方法当前所属的类中获取拟被调用虚方法当前所属的类的引用,判断拟被调用虚方法当前所属的类的引用的最高位是否为第二预设值,如果是,执行步骤C2,否则,执行步骤C3;
- [0216] 可选地,本实施例中,每一个虚方法的长度为4个字节,其中,第一个字节为虚方法的实例标签,第二个字节和第三个字节为虚方法所属的类的引用,第四个字节为虚方法的索引,其中,虚方法的实例标签用于表示虚方法的实例的类型;
- [0217] 相应地,Java卡从拟被调用虚方法中获取拟被调用虚方法当前所属的类的引用,具体为:Java卡从拟被调用虚方法中获取第二个字节和第三个字节上的数据得到拟被调用虚方法当前所属的类的引用;
- [0218] 可选地,本实施例中,第二预设值具体为1;
- [0219] 例如,本实施例中,拟被调用虚方法当前所属的类的引用具体为:80 03,其最高位为1;
- [0220] 步骤C2:Java卡从拟被调用虚方法当前所属的类中获取拟被调用虚方法当前所属

的类的父类的引用,从拟被调用虚方法当前所属的类的父类的引用中分别获取拟被调用虚方法当前所属的类的索引和拟被调用虚方法所属的包的索引,根据拟被调用虚方法所属的包的索引获取对应的导出表,根据拟被调用虚方法当前所属的类的索引在导出表中获取拟被调用虚方法当前所属的类的地址;

[0221] 可选地,本实施例中,一个虚方法所属的类的引用的长度为两个字节,即包含16位,其中第0位至第7位是该虚方法所属的类的索引,为低位字节,第8位至第14位是该虚方法所属的包的索引,为高位字节,第15位为最高位字节;

[0222] 可选地,本实施例中,Java卡从拟被调用虚方法当前所属的类的父类的引用中分别获取拟被调用虚方法当前所属的类的索引和拟被调用虚方法所属的包的索引,具体包括:Java卡从拟被调用虚方法当前所属的类的父类的引用中获取第0位至第7位得到拟被调用虚方法当前所属的类的索引,获取第8位至第14位得到拟被调用虚方法所属的包的索引;

[0223] 例如,本实施例中,拟被调用虚方法当前所属的类的索引具体为:03,拟被调用虚方法所属的包的索引具体为:00;

[0224] 可选地,本实施例中,根据拟被调用虚方法所属的包的索引获取对应的导出表,具体为:Java卡根据拟被调用虚方法所属的包的索引在预存的导出表地址数组中获取对应的导出表地址,根据导出表地址获取拟被调用虚方法所属的包的索引对应的导出表;

[0225] 例如,本实施例中,拟被调用虚方法所属的包的索引具体为:00;

[0226] 导出表地址数组具体为:

[0227] 00 00 1A 87

[0228] 00 00 1E 06

[0229] 00 00 20 32

[0230] 00 00 22 87

[0231] 00 00 43 02

[0232] 00 00 45 34;

[0233] 导出表地址具体为:00 00 1A 87;

[0234] 导出表具体为:

[0235]

[0236] 0A

[0237] 01 07 //size

[0238] 0D //class_count

[0239] //第0个

[0240] 00 E2 //class_offset

[0241] 00 01 //static_field_count

[0242] 00 01 //static_method_count

[0243] //第1个

[0244] 01 D2 //class_offset

[0245] 00 02 //static_field_count

[0246] 00 03 //static_method_count

[0247] //第2个

[0248] 01 FE //class_offset
[0249] 00 0A //static_field_count
[0250] 00 03 //static_method_count
[0251] //第3个
[0252] 03 02 //class_offset
[0253] 00 03 //static_field_count
[0254] 00 08 //static_method_count
[0255]

[0256] 例如,本实施例中,拟被调用虚方法当前所属的类的索引为03,以03为索引在导出表中可以找到拟被调用虚方法当前所属的类的地址偏移为03 02;

[0257] 步骤C3:Java卡从拟被调用虚方法所属的包信息中获取拟被调用虚方法所属的包的类组件的基地址,根据拟被调用虚方法所属的包的类组件的基地址和拟被调用虚方法当前所属的类的地址偏移得到拟被调用虚方法当前所属的类的地址;

[0258] 可选地,本实施例中,根据拟被调用虚方法所属的包的类组件的基地址和拟被调用虚方法当前所属的类的地址偏移得到拟被调用虚方法当前所属的类的地址,具体为:Java卡从拟被调用虚方法所属的包信息中获取拟被调用虚方法所属的包的类组件的基地址,将拟被调用虚方法所属的包的类组件的基地址和拟被调用虚方法当前所属的类的地址偏移进行加法运算得到第一运算值,将第一运算值作为拟被调用虚方法当前所属的类的地址;

[0259] 例如:拟被调用虚方法所属的包的类组件的基地址具体为:00 00 02 50,拟被调用虚方法当前所属的类的地址具体为:00 00 05 52;

[0260] 步骤303:Java卡从虚拟机运行栈中获取对象引用,以对象引用为索引在保存的对象表中获取对象数据;

[0261] 可选地,本实施例中,Java卡从虚拟机运行栈中获取对象引用,具体为:Java卡获取保存的栈指针,以栈指针为地址获取栈指针指向的数据,从栈指针指向的数据中获取前两个字节上的数据得到对象引用;

[0262] 例如,本实施例中,栈指针指向的数据具体为:00 8f 00 03 00 00 fc 10 00 01,对象引用具体为:00 8f;

[0263] 可选地,本实施例中,以对象引用为索引在保存的对象表中获取对象数据,具体包括:

[0264] 步骤B1:Java卡获取对象数据的基地址、对象数据的起始编号和对象数据地址的最大长度,将对象引用与对象数据的起始编号做减法运算得到第二运算值,将第二运算值与对象数据地址的最大长度做乘法运算得到第三运算值,将第三运算值与对象数据的基地址做加法运算得到对象数据的地址;

[0265] 例如,本实施例中,对象数据的基地址具体为:00 20 00 00;

[0266] 对象数据的起始编号具体为:00 80;

[0267] 对象数据地址的最大长度具体为:4;

[0268] 对象引用具体为:008f

[0269] 得到的对象数据的地址具体为:0x0020003C;

[0270] 步骤B2:Java卡获取对象数据的地址对应的数据得到对象数据;

[0271] 例如,本实施例中,对象数据具体为:20 01 0C 01 43 18 00 00 fc 10 a1 00 00 00 0a 1b 00 08 2d 05 81 00 00 00 00 1b 00 02 2d 15 91 00 00 00 00 1b 00 85 2d 17 20 00 00 01 f5 05 fc 0d 00 8f 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 20 00 00 00 b2 05 81 00 00 00 00 1b 00 02 2d 9c 28 00 00 00 fe 07 fc 0c 81 00 00 00 00 1b 00 02 2d 9e 28 00 00 02 8a 05 fc 0b 81 00;

[0272] 步骤304:Java卡从对象数据中获取对象实际类所属的包的标识和对象实际类的引用,根据对象实际类所属的包的标识和对象实际类的引用得到对象实际类的地址;

[0273] 可选地,本实施例中,Java卡从对象数据中获取对象实际类所属的包的标识和对象实际类的引用,具体为:Java卡从对象数据中获取第四个字节上的数据得到对象实际类所属的包的标识,获取第二个字节和第三个字节上的数据得到对象实际类的引用;

[0274] 例如,本实施例中,对象实际类所属的包的标识具体为:01;

[0275] 对象实际类的引用具体为:010C;

[0276] 可选地,本实施例中,根据对象实际类所属的包的标识和对象实际类的引用得到对象实际类的地址,具体包括:

[0277] 步骤D1:Java卡以对象实际类所属的包的标识为索引在包信息地址列表中获取对象实际类所属的包地址信息,获取对象实际类所属的包地址信息对应的数据得到对象实际类所属的包信息,从对象实际类所属的包信息中获取前四个字节上的数据得到对象实际类所属的包的类组件的起始地址;

[0278] 例如,本实施例中,对象实际类所属的包的地址信息具体为:00 01 20 00;

[0279] 对象实际类所属的包信息具体为:00 00 62 12 00 00 12 3b 00 00 62 64 00 00 12 8d 00 00 62 b6 00 00 12 df 00 00 63 08 00 00 13 31 00 00 63 5a 00 00 13 83;

[0280] 对象实际类所属的包的类组件的起始地址具体为:00 00 62 12;

[0281] 步骤D2:Java卡将对象实际类所属的包的类组件的起始地址和对象实际类的引用进行加法运算得到对象实际类的地址;

[0282] 例如,本实施例中,对象实际类的地址具体为:0000631E;

[0283] 步骤305:Java卡判断拟被调用虚方法当前所属的类的地址与对象实际类的地址是否相同,如果是,执行步骤312,否则,执行步骤306;

[0284] 例如,本实施例中,拟被调用虚方法当前所属的类的地址具体为:00 00 05 52,对象实际类的地址具体为:0000631E,二者不同,执行步骤306;

[0285] 步骤306:Java卡根据拟被调用虚方法当前所属的类的地址和对象实际类的地址建立链表;

[0286] 具体地,本实施例中,步骤306具体为:

[0287] 步骤M1:Java卡建立链表;

[0288] 步骤M2:Java卡将对象实际类的地址添加到链表中;

[0289] 步骤M3:Java卡判断拟被调用虚方法当前所属的类的地址和对象实际类的地址是否相同,如果是,链表建立完成,否则,执行步骤M4;

[0290] 步骤M4:Java卡将对象实际类的地址设置为对象实际类的父类的地址,返回步骤

M2;

[0291] 可选地,本实施例中,链表的头为拟被调用虚方法当前所属的类的地址,链表的尾为对象实际类的地址,链表中的每一个节点代表一个类的地址;

[0292] 例如,本实施例中,链表具体为:

[0293] 06 00 00 01 00 00 80 00 00 FF 00

[0294] 01 /*base*/

[0295] 04 /*count*/

[0296] 00 08

[0297] 00 0C

[0298] 00 10

[0299] 00 15

[0300] 00 FF FF FF FF /*已经创建好的VMMT*/

[0301] 03

[0302] 06 80 03 02 00 00 80 00 00 FF 00

[0303] 03 /*base*/

[0304] 04 /*count*/

[0305] 00 10

[0306] 00 15

[0307] 00 1A

[0308] 00 2C

[0309] 00 01 02 03 04 FF FF /*已经创建好的VMMT*/

[0310] 04

[0311] 在本例如中,对象实际类是拟被调用虚方法当前所属的类的直接子类,因此,链表中只有对象实际类的地址和拟被调用虚方法当前所属的类的地址两项;

[0312] 步骤307:Java卡根据拟被调用虚方法当前所属的类的地址判断链表中是否存在拟被调用虚方法当前所属的类的直接子类的地址,如果是,执行步骤308,否则,执行步骤310;

[0313] 可选地,本实施例中,Java卡根据拟被调用虚方法当前所属的类的地址判断链表中是否存在拟被调用虚方法当前所属的类的直接子类的地址,具体为:Java卡根据拟被调用虚方法当前所属的类的地址在链表中查找拟被调用虚方法当前所属的类所在的节点,判断拟被调用虚方法当前所属的类所在的节点之后是否还存在节点,如果是,判定链表中存在拟被调用虚方法当前所属的类的直接子类的地址,否则,判定链表中不存在拟被调用虚方法当前所属的类的直接子类的地址;

[0314] 例如,本实施例中,拟被调用虚方法当前所属的类存在直接子类,其直接子类为对象实际类,执行步骤308;

[0315] 步骤308:Java卡根据拟被调用虚方法当前所属的类的直接子类的地址获取拟被调用虚方法当前所属的类的直接子类,在拟被调用虚方法当前所属的类的直接子类的虚方法映射表中查找拟被调用虚方法的索引,判断是否查找到,如果是,执行步骤309,否则,执行步骤310;

[0316] 可选地,本实施例中,步骤308之前还包括:Java卡从拟被调用虚方法当前所属的类中获取拟被调用虚方法所属的类的虚方法的个数和拟被调用虚方法所属的类的虚方法的起始编号并保存;

[0317] 例如,本实施例中,拟被调用虚方法的索引具体为:02,其直接子类的虚方法映射表为00 01 02 03 04 FF FF,可以查找到拟被调用虚方法的索引02,执行步骤309;

[0318] 步骤309:Java卡将拟被调用虚方法当前所属的类的直接子类的地址设置为拟被调用虚方法当前所属的类的地址,将拟被调用虚方法当前所属的类在其直接子类的虚方法映射表中的索引设置为拟被调用虚方法当前所属的类的索引,返回步骤307;

[0319] 例如,本实施例中,返回步骤307之后,拟被调用虚方法当前所属的类不存在直接子类,执行步骤310;

[0320] 步骤310:Java卡在拟被调用虚方法当前所属的类的虚方法表中查找拟被调用虚方法的索引,判断是否查找到,如果是,执行步骤312,否则,执行步骤311;

[0321] 可选地,本实施例中,步骤310具体为:Java卡根据拟被调用虚方法所属的类的虚方法的个数和拟被调用虚方法所属的类的虚方法的起始编号得到第四运算值,判断拟被调用虚方法的索引是否大于或者等于拟被调用虚方法所属的类的虚方法的起始编号且小于第四运算值,如果是,执行步骤312,否则,执行步骤311;

[0322] 可选地,本实施例中,Java卡根据拟被调用虚方法所属的类的虚方法的个数和拟被调用虚方法所属的类的虚方法的起始编号得到第四运算值,具体为:Java卡将拟被调用虚方法所属的类的虚方法的个数和拟被调用虚方法所属的类的虚方法的起始编号进行加法运算得到第四运算值;

[0323] 例如,本实施例中,拟被调用虚方法所属的类的虚方法的个数具体为:07,拟被调用虚方法所属的类的虚方法的起始编号具体为:03,拟被调用虚方法的索引具体为:02,第四运算值具体为:10,本实施例中,拟被调用虚方法的索引02不在03和10 之间,执行步骤311;

[0324] 步骤311:Java卡从拟被调用虚方法当前所属的类中获取拟被调用虚方法当前所属的类的父类的引用,根据拟被调用虚方法当前所属的类的父类的引用确定拟被调用虚方法当前所属的类的地址,返回步骤310;

[0325] 可选地,本实施例中,步骤311具体包括:

[0326] 步骤N1:Java卡从拟被调用虚方法当前所属的类中获取拟被调用虚方法当前所属的类的父类的引用,判断拟被调用虚方法当前所属的类的父类的引用的最高位是否为第二预设值,如果是,执行步骤N2,否则,执行步骤N3;

[0327] 可选地,本实施例中,第二预设值具体为1;

[0328] 例如,本实施例中,拟被调用虚方法当前所属的类的父类的引用具体为:8003;

[0329] 步骤N2:Java卡从拟被调用虚方法当前所属的类的父类的引用中获取拟被调用虚方法当前所属的类的父类的索引和拟被调用虚方法当前所属的类的父类所属的包的索引并分别作为拟被调用虚方法当前所属的类的索引和拟被调用虚方法所属的包的索引,根据所述拟被调用虚方法所属的包的索引获取对应的导出表,根据所述拟被调用虚方法当前所属的类的索引在所述导出表中获取拟被调用虚方法当前所属的类的地址;

[0330] 可选地,本实施例中,Java卡从拟被调用虚方法当前所属的类的父类的引用中获

取拟被调用虚方法当前所属的类的父类的索引和拟被调用虚方法当前所属的类的父类所属的包的索引并分别作为拟被调用虚方法当前所属的类的索引和拟被调用虚方法所属的包的索引,具体为:Java卡从拟被调用虚方法当前所属的类的父类的引用中获取的第0位至第7位作为拟被调用虚方法当前所属的类的索引,获取第8位至第14位作为拟被调用虚方法所属的包的索引,返回步骤310;

[0331] 步骤N3:Java卡从拟被调用虚方法所属的包信息中获取拟被调用虚方法所属的包的类组件的基地址,根据拟被调用虚方法所属的包的类组件的基地址和拟被调用虚方法当前所属的类的地址偏移得到拟被调用虚方法当前所属的类的地址,返回步骤310;

[0332] 例如,本实施例中,返回步骤310之后,拟被调用虚方法所在的当前类的虚方法的个数具体为:04,拟被调用虚方法所在的虚方法映射表中虚方法的起始编号具体为:01,拟被调用虚方法的索引具体为:02,第四运算值具体为:05,本实施例中,拟被调用虚方法的索引02在01和05 之间,执行步骤312;

[0333] 步骤312: Java卡根据拟被调用虚方法的索引从拟被调用虚方法当前所属的类的虚方法中获取实际被调用虚方法的地址。

[0334] 可选地,本实施例中,步骤312具体为:Java卡根据拟被调用虚方法的索引和拟被调用虚方法所属的类的虚方法的起始编号进行减法运算得到第五运算值,以第五运算值为索引在拟被调用虚方法当前所属的类的虚方法表中获取实际被调用虚方法的地址。

[0335] 例如,本实施例中,被调用虚方法的索引具体为:02,拟被调用虚方法所属的类的虚方法的起始编号具体为:01,第五运算值具体为:01;

[0336] 拟被调用虚方法当前所属的类的虚方法表具体为:

[0337] 06 00 00 01 00 00 80 00 00 FF 00

[0338] 01 /*base*/

[0339] 04 /*count*/

[0340] 00 08 //第0个

[0341] 00 0C //第1个

[0342] 00 10//第2个

[0343] 00 15//第3个

[0344] 00 FF FF FF FF /*已经创建好的VMMT*/

[0345] 被调用虚方法具体为:00 0C;

[0346] 本实施例提供了一种虚方法的调用方法,通过虚方法映射表来索引原应用和新应用中的虚方法,在父类所属的应用文件包升级后,可以使子类所属的应用文件包无需重新生成Java字节码仍可正常运行,使得虚方法映射表既可以服务原应用又可以服务新应用,提高了Java卡的API的扩展性。

[0347] 本发明还提供了一种电子设备,如图4所示,其示出了适于用来实现本申请各实施例的电子设备(例如图1中的Java卡)400的结构示意图。本申请实施例中的设备可以包括但不限于诸如移动电话、笔记本电脑、数字广播接收器、PDA(个人数字助理)、PAD(平板电脑)、PMP(便携式多媒体播放器)、车载终端(例如车载导航终端)等等的移动终端以及诸如数字TV、台式计算机等等的固定终端。图4示出的电子设备仅仅是一个示例,不应对本申请实施例的功能和使用范围带来任何限制。

[0348] 如图4所示,设备400可以包括处理装置(例如中央处理器、图形处理器等)401,其可以根据存储在只读存储器(ROM)402中的程序或者从存储装置408加载到随机访问存储器(RAM)403中的程序而执行各种适当的动作和处理。在RAM 403中,还存储有电子设备300操作所需的各种程序和数据。处理装置401、ROM 402以及RAM 403通过总线404彼此相连。输入/输出(I/O)接口405也连接至总线304。

[0349] 通常,以下装置可以连接至I/O接口405:包括例如触摸屏、触摸板、键盘、鼠标、摄像头、麦克风、加速度计、陀螺仪等的输入装置406;包括例如液晶显示器(LCD)、扬声器、振动器等的输出装置407;包括例如磁带、硬盘等的存储装置408;以及通信装置409。通信装置409可以允许设备400与其他设备进行无线或有线通信以交换数据。虽然图4示出了具有各种装置的设备400,但是应理解的是,并不要求实施或具备所有示出的装置。可以替代地实施或具备更多或更少的装置。

[0350] 特别地,根据本申请的实施例,上文参考流程图描述的过程可以被实现为计算机软件程序。例如,本申请的实施例包括一种计算机程序产品,其包括承载在计算机可读介质上的计算机程序,该计算机程序包含用于执行流程图所示的方法的程序代码。在这样的实施例中,该计算机程序可以通过通信装置409从网络上被下载和安装,或者从存储装置408被安装,或者从ROM 402被安装。在该计算机程序被处理装置401执行时,执行本申请实施例的方法中限定的上述功能。

[0351] 需要说明的是,本申请上述的计算机可读介质可以是计算机可读信号介质或者计算机可读存储介质或者是上述两者的任意组合。计算机可读存储介质例如可以是一——但不限于——电、磁、光、电磁、红外线、或半导体的系统、装置或器件,或者任意以上的组合。计算机可读存储介质的更具体的例子可以包括但不限于:具有一个或多个导线的电连接、便携式计算机磁盘、硬盘、随机访问存储器(RAM)、只读存储器(ROM)、可擦式可编程只读存储器(EPROM或闪存)、光纤、便携式紧凑磁盘只读存储器(CD-ROM)、光存储器件、磁存储器件、或者上述的任意合适的组合。在本申请中,计算机可读存储介质可以是任何包含或存储程序的有形介质,该程序可以被指令执行系统、装置或者器件使用或者与其结合使用。而在本申请中,计算机可读信号介质可以包括在基带中或者作为载波一部分传播的数据信号,其中承载了计算机可读的程序代码。这种传播的数据信号可以采用多种形式,包括但不限于电磁信号、光信号或上述的任意合适的组合。计算机可读信号介质还可以是计算机可读存储介质以外的任何计算机可读介质,该计算机可读信号介质可以发送、传播或者传输用于由指令执行系统、装置或者器件使用或者与其结合使用的程序。计算机可读介质上包含的程序代码可以用任何适当的介质传输,包括但不限于:电线、光缆、RF(射频)等等,或者上述的任意合适的组合。

[0352] 上述计算机可读介质可以是上述设备中所包含的;也可以是单独存在,而未装配入该设备中。

[0353] 上述计算机可读介质承载有一个或者多个程序,当上述一个或者多个程序被该设备执行时,使得该设备可以执行上述实施例中的虚方法的调用方法。

[0354] 可以以一种或多种程序设计语言或其组合来编写用于执行本申请的操作的计算机程序代码,上述程序设计语言包括面向对象的程序设计语言—诸如Java、Smalltalk、C+++,还包括常规的过程式程序设计语言—诸如“C”语言或类似的程序设计语言。程序代码可

以完全地在用户计算机上执行、部分地在用户计算机上执行、作为一个独立的软件包执行、部分在用户计算机上部分在远程计算机上执行、或者完全在远程计算机或服务器上执行。在涉及远程计算机的情形中,远程计算机可以通过任意种类的网络——包括局域网(LAN)或广域网(WAN)——连接到用户计算机,或者,可以连接到外部计算机(例如利用因特网服务提供商来通过因特网连接)。

[0355] 附图中的流程图和框图,图示了按照本申请各种实施例的系统、方法和计算机程序产品的可能实现的体系架构、功能和操作。在这点上,流程图或框图中的每个方框可以代表一个模块、程序段、或代码的一部分,该模块、程序段、或代码的一部分包含一个或多个用于实现规定的逻辑功能的可执行指令。也应当注意,在有些作为替换的实现中,方框中所标注的功能也可以以不同于附图中所标注的顺序发生。例如,两个接连地表示的方框实际上可以基本并行地执行,它们有时也可以按相反的顺序执行,这依所涉及的功能而定。也要注意,框图和/或流程图中的每个方框、以及框图和/或流程图中的方框的组合,可以用执行规定的功能或操作的专用的基于硬件的系统来实现,或者可以用专用硬件与计算机指令的组合来实现。

[0356] 描述于本申请实施例中所涉及到的单元可以通过软件的方式实现,也可以通过硬件的方式来实现。其中,单元的名称在某种情况下并不构成对该单元本身的限定。

[0357] 本申请提供的电子设备,适用于本申请的虚方法的调用方法的任一实施例,在此不再赘述。

[0358] 本发明提供了一种电子设备,通过虚方法映射表来索引原应用和新应用中的虚方法,在父类所属的应用文件包升级后,可以使子类所属的应用文件包无需重新生成Java字节码仍可正常运行,使得虚方法映射表既可以服务原应用又可以服务新应用,提高了Java卡的API的扩展性。

[0359] 本申请提供了一种计算机可读存储介质,该计算机可读存储介质存储计算机指令,计算机指令使计算机执行上述实施例所示的虚方法的调用方法。

[0360] 本申请提供的计算机可读存储介质,适用于上述的调用虚方法的任一实施例,在此不再赘述。

[0361] 本申请提供了一种计算机程序产品,当计算机程序产品在计算机上运行时,使得计算机执行本申请中的虚方法的调用方法。

[0362] 以上所述,仅为本发明较佳的具体实施方式,但本发明的保护范围并不局限于此,任何熟悉本技术领域的技术人员在本发明揭露的技术范围内,可轻易想到变化或替换,都应涵盖在本发明的保护范围之内。因此,本发明的保护范围应以权利要求的保护范围为准。

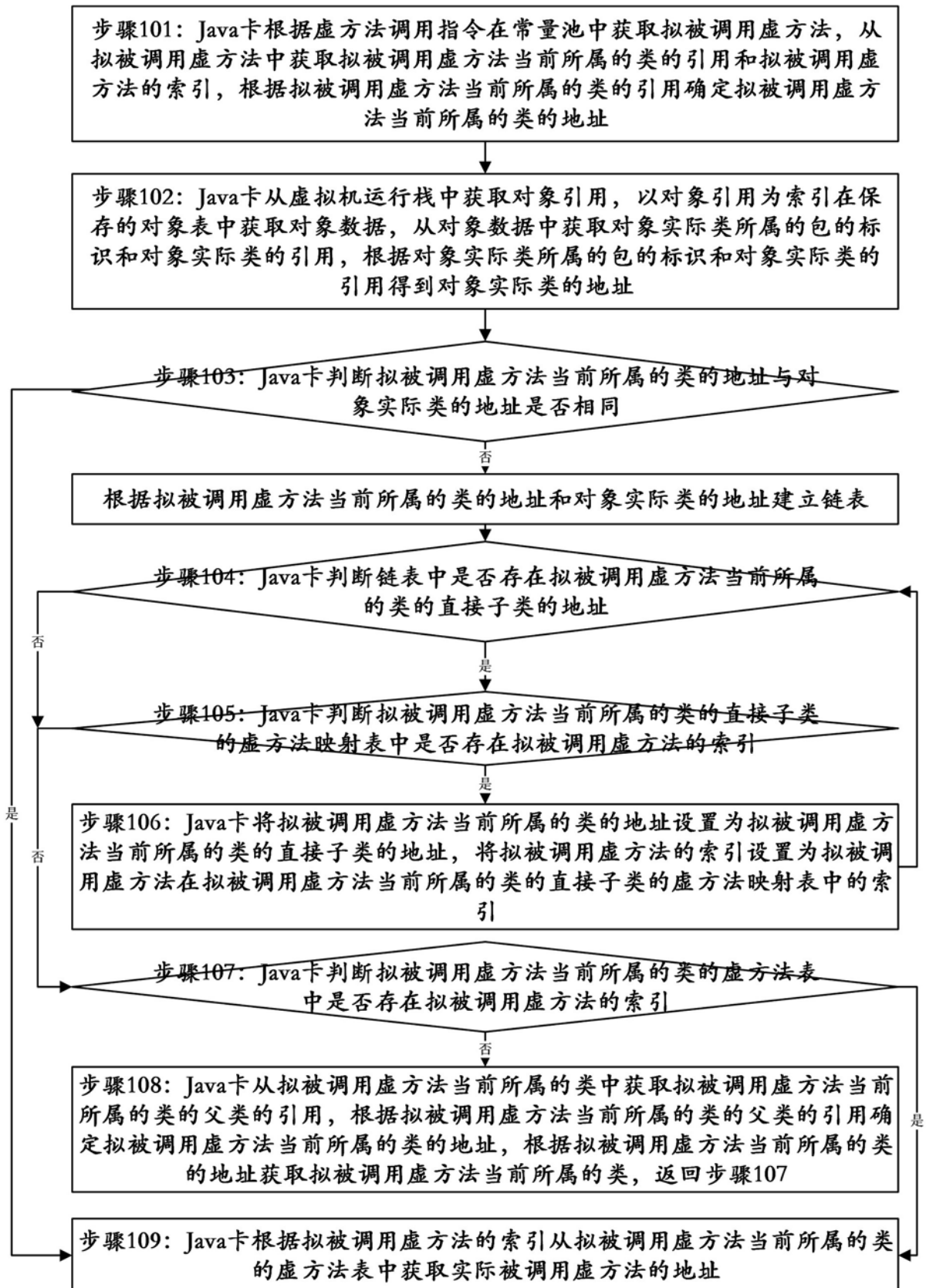


图1

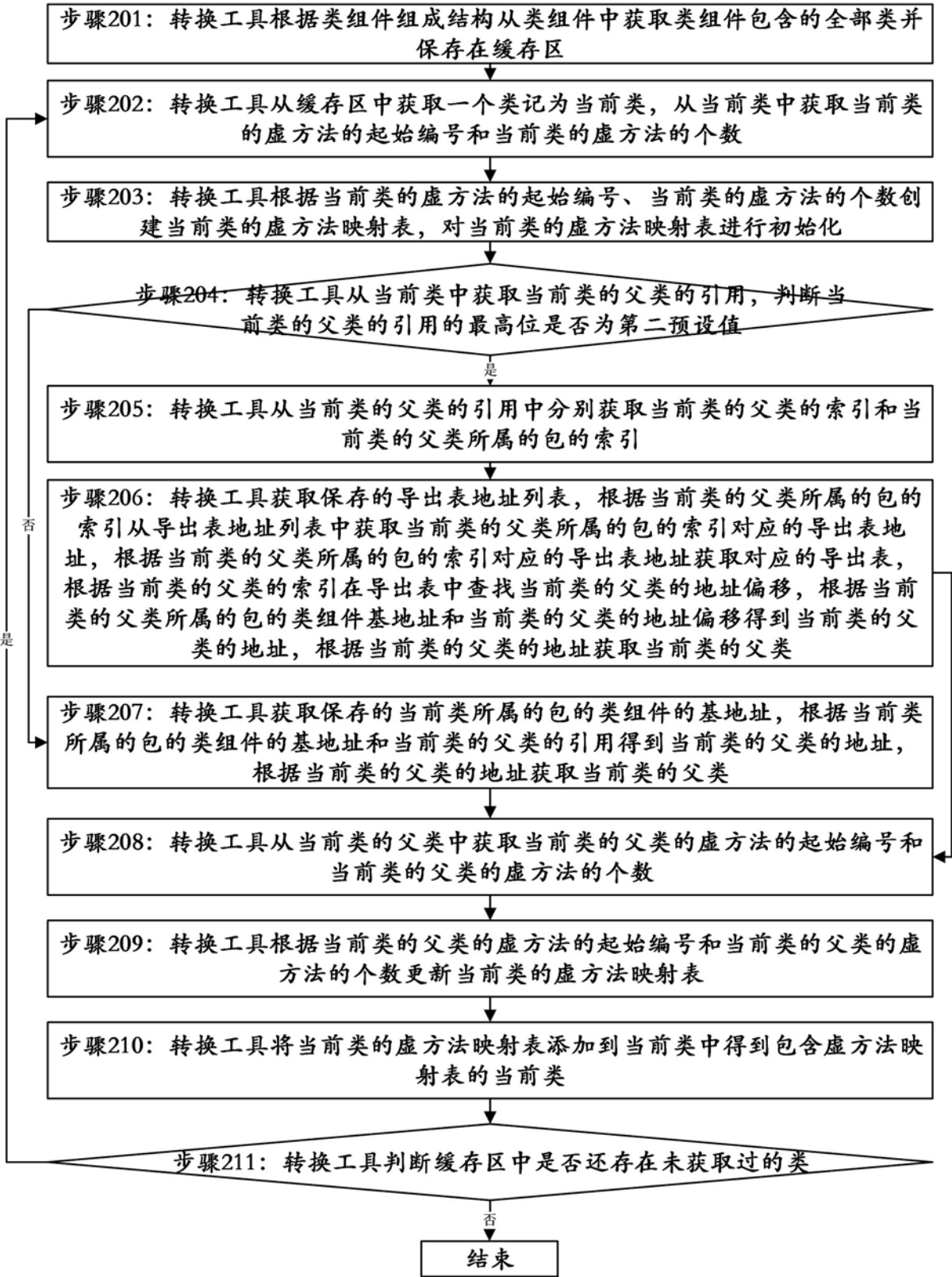


图2

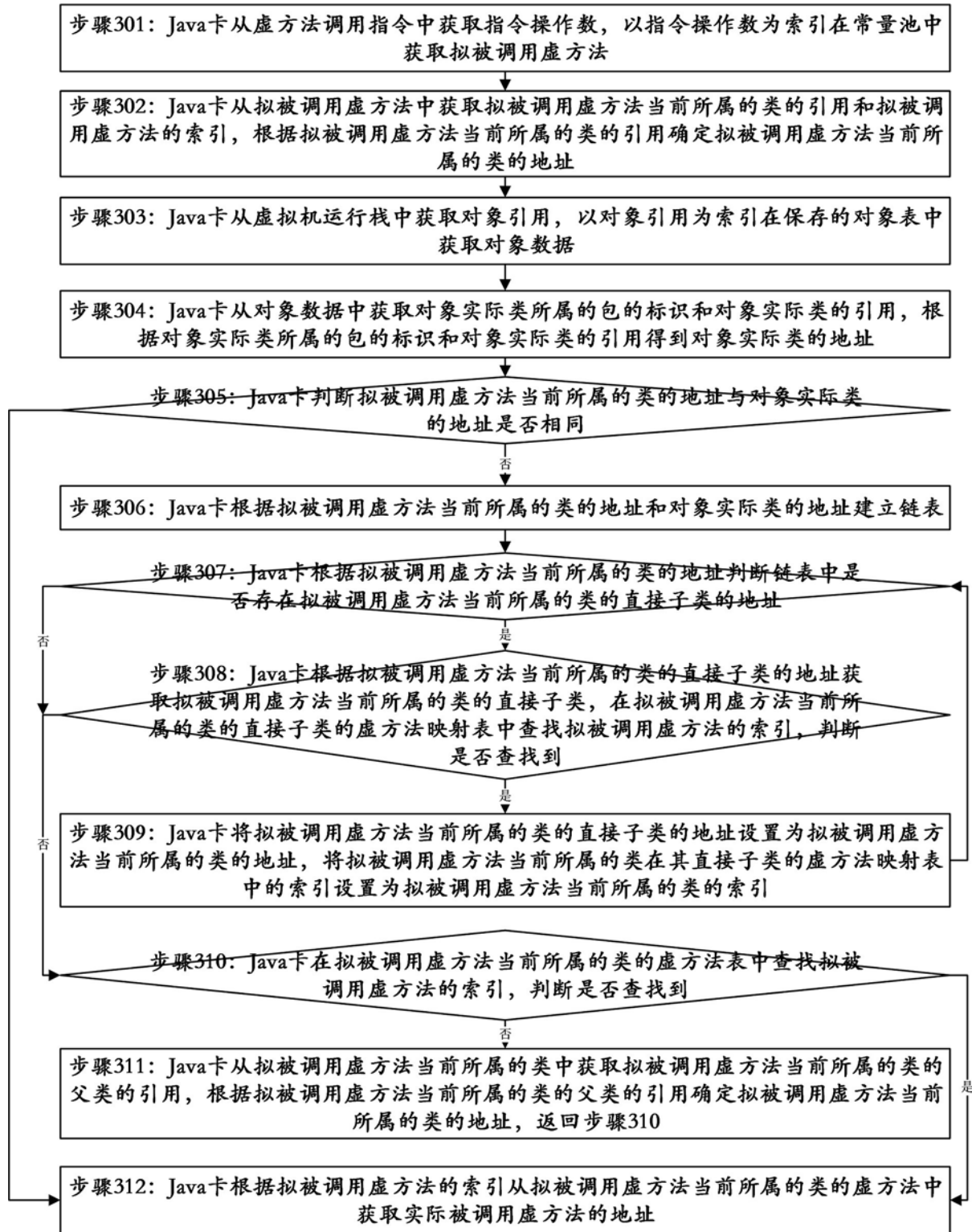


图3

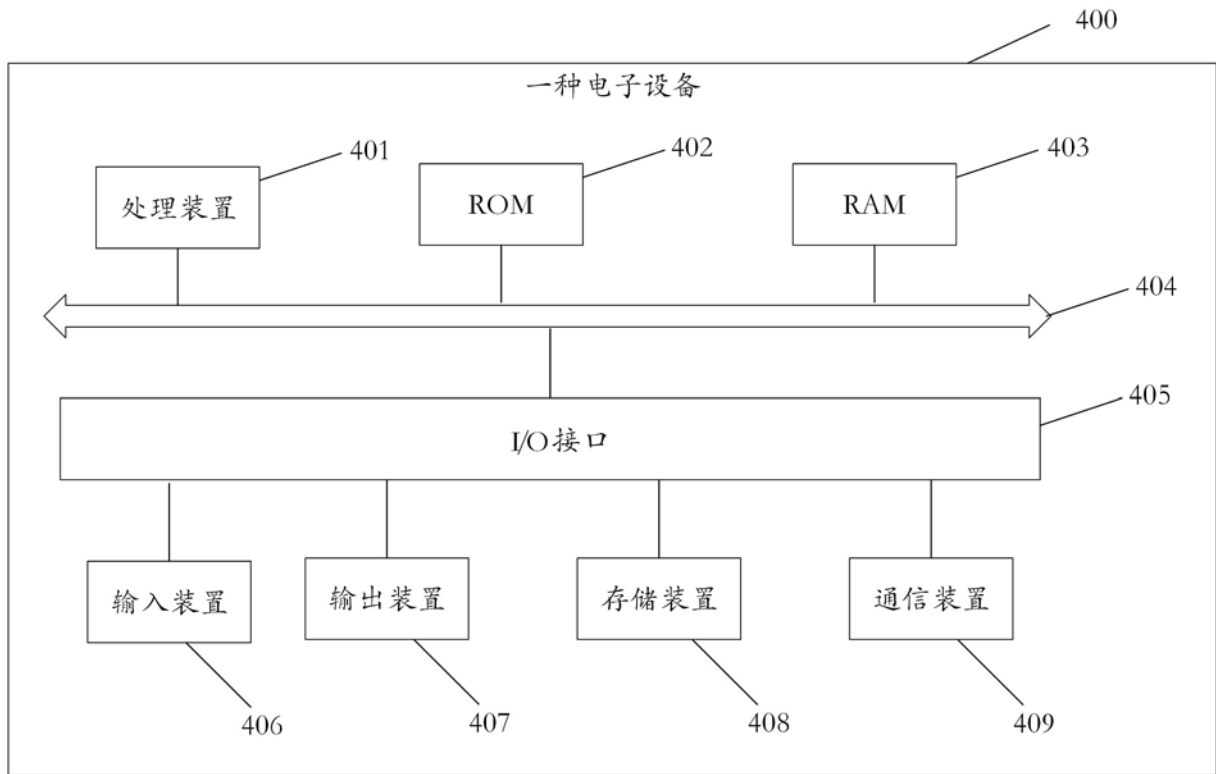


图4