[19]中华人民共和国专利局

[51]Int.Cl[6]
G06F 17/00

[12] 发明专利申请公开说明书

[21] 申请号 96121868.1

[43]公开日 1997 年 9 月 3 日

[11] 公开号 CN 1158459A

[22]申请日 96.12.6

[30]优先权

[32]95.12.8 [33]US[31]569754

[71]申请人 太阳微系统有限公司

地址 美国加利福尼亚州

[72]发明人 弗兰克·耶林

[74]专利代理机构 柳沈知识产权律师事务所

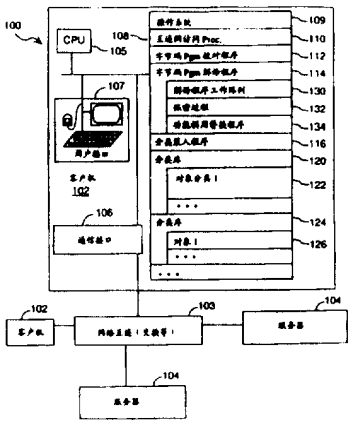
代理人 孙履平

权利要求书 3 页 说明书 9 页 附图页数 4 页

[54]发明名称 在保密解译程序中优化运行时调用专用变量函数的系统和方法

[57]摘要

一个保密程序解译程序首先执行一个特殊检查，它执行一个方法调用，以确定调用方法的本来目的是对一个专用变量进行访问、还是对所述专用变量值进行修改或是回送一个恒定值，当是这种情况时，正在被执行方法的解译程序的内部表示被修改，以直接访问被调用方法的专用变量，或直接访问被调用方法的所存储的恒定值。修改后的方法表示使用在一般的源代码程序中没有的多个特殊的“特许”装载和存储指令，用于访问正在被执行方法以外的专用变量和恒定值而不会被标记保密违反。



权 利 要 求 书

1、一种计算机系统，包括:

存储器，用于存储多个对象和多个过程，所述对象中的每一个都包括一个相关对象类的事例和所述过程中的每一个都属于相应的对象类，所述多个过程包括多个简单的过程，其中，由每个所述简单过程执行的整个功能从如下组成的组中进行选择: (A)回送一个专用变量值，其中，所述的专用变量被存储并专用于所述简单过程所属的所述对象类中的一个对象中，(B)把一个规定值存储到所述的专用变量中，(C)回送一个恒定值；和

保密程序解译程序，用于执行所述多个过程中被选择的一些过程，所述解译程序包括一个装载子程序，用于产生将被执行的所述过程中第一个过程的工作表达，还包括一个优化子程序，用于当被另一个所述过程调用时，优化所述简单过程的执行，当所述解译程序在处理从所述多个过程的第一个过程到第二个过程的过程调用时，所述优化子过程确定所述第二过程是否所述简单过程中的一个，和当所述的确定是肯定的时，用一个直接访问指令替换在所述第一过程的所述工作表达中的所述过程调用，其中，所述直接访问指令是从如下组成的组中选择: (A)用于直接回送所述专用变量值的第一指令，(B)用于将一个规定值直接存储到所述专用变量中的第二指令，和(C)用于直接存储所述恒定值的第三指令。

2、如权利要求 1 所述的计算机系统，其中，

所述解译程序包括多个保密指令，用于避免将一个变量值装入到一个操作栈并将一个操作栈上的值存储到一个变量上的多个标准指令访问没有存储在驻留所述多个标准指令的过程的对象类的一个对象中的任意一个专用变量，和

所述第一和第二指令是特殊目的指令，即使在所述第一过程和第二过程属于不同的各自对象类的情况下，这些指令也能够对所述的专用变量进行访问，而不会被所述的多个保密指令标记一个保密违反。

3、如权利要求 2 所述的计算机系统，其中，所述的解译程序包括多个指令，用于当所述第一过程执行结束时，从所述的程序解译程序过程中清洗所述第一过程的所述工作表达，由此，当再次执行所述的第一过程时，所述解译程序产生所述第一过程的一个新的工作表达。

4、一种操作计算机系统的方法，包括如下步骤:

在计算机存储器中存储多个对象和多个过程，每个所述的对象包括相关对象类的一个事例和每个所述的过程属于一个相应的对象类，所述多个过程包括多个简单的过程，其中由每个所述简单过程执行的全部功能是根据如下组成的组进行选择的: (A)回送一个专用变量值，其中，所述的专用变量专用于所述简单过程所属的所述对象类的一个对象，(B)将一个规定值存储到所述的专用变量中，和(C)回送一个恒定值;

在源程序解译程序的控制下，执行所述多个过程中所选择的一个，包括产生将被执行的所述多个过程中第一个过程的一个工作表达，和当由所述第一过程调用时优化所述多个简单过程中任意一个的执行，所述的优化步骤包括当所述解译程序在处理从所述多个过程的第一个过程到第二个过程的过程调用时，确定所述的第二过程是否所述多个简单程序中的一个，和当所述的确定结果是肯定的时，利用一个直接访问指令置换在所述第一过程的所述工作表达中的所述过程调用，其中，所述直接访问指令是从如下组成的组中选择的: (A)一个用于直接回送所述专用变量值的第一指令，(B)一个用于直接将一个规定值存储到所述专用变量中的第二指令，和(C)一个用于直接存储所述恒定值的第三指令。

5、如权利要求4所述的方法，其中，所述的第一和第二指令是特殊目的指令，所述的方法包括如下步骤:

避免用于将一个变量值装入到一个操作栈并用于把所述操作栈上的一个值存储到一个变量上的标准指令访问驻留在所述标准指令的过程以外的任意一个专用变量，和当任意一个标准指令的执行需要对任意一个专用变量进行访问、而该专用变量并没有存储在驻留有所述标准指令的过程的所述对象类的一个对象中时，标记一个保密违反；和

允许所述第一和第二指令去访问所述的专用变量，并且即使当所述第一过程和所述第二过程属于不同的各自对象类时也不会被标记一个保密违反。

6、如权利要求5所述的方法，包括:

当所述第一过程的执行结束时从所述解译程序中清洗所述第一过程的所述工作表达，以使当再次执行所述的第一过程时，所述的解译程序将产生一个新的所述第一过程的工作表达。

7、一种存储器，用于存储由一个数据处理系统正在被执行的多个程序

访问的数据，所述的存储器包括:

存储在所述存储器中的多个对象和多个过程，每个所述对象包括一个相关对象类的事例和每个所述过程属于一个相应的对象类，所述多个过程包括多个简单过程，其中，由每个简单过程执行的全部功能是从如下组成的组中选择的: (A)回送一个专用变量值，其中，所述的专用变量值被存储并专用于所述简单过程所属的对象类的一个对象中; (B)把一个规定值存储到所述专用变量中; 和(C)回送一个标准值; 和

存储在所述存储器中的一个保密程序解译程序，用于执行从所述多个过程中所选择的过程，所述解译程序包括一个装载子程序，用于产生将被执行的所述多个过程中第一个过程的一个工作表达，并包括一个优化子过程，用于当被所述多个过程的另外一些过程调用时，优化所述多个简单过程的执行，当所述解译程序正在处理从所述多个过程的第一过程到第二过程的过程调用时，所述的优化子过程确定所述第二字过程是否所述多个简单过程中的一个，如果所述确定的结果是肯定的，则利用一个直接访问指令置换在所述第一过程的所述工作表达中的所述过程调用，其中，所述的直接访问指令是从如下组成的组中选择的: (A)用于直接回送所述专用变量值的第一指令, (B)用于直接将一个规定值存储到所述专用变量中的第二指令, 和(C)用于直接存储所恒定值的第三指令。

8、如权利要求 7 所述的存储器，其中，

所述的解译程序包括多个保密指令，用于避免将一个专用变量装入到一个操作栈并将在操作栈上的一个值存储到一个变量中的多个标准指令访问任意一个专用变量，而该专用变量并没有存储在驻留有所述多个标准指令的过程的所述对象类的一个对象中; 和

其中，所述的第一和第二指令是特殊目的指令，它们能够访问所述专用变量，并且即使在第一过程和所述第二过程属于不同的各自对象类时也不会引起被所述保密指令标记一个保密违反。

9、如权利要求 8 所述的存储器，其中，

所述的解译程序包括多个指令，用于当所述第一过程的执行结束时，从所述的过程解译程序中清洗所述第一过程的所述工作表达，以使当再次执行所述第一过程时，所述解译程序产生一个新的所述第一过程的工作表达。

说　明　书

在保密解译程序中优化运行时调用专用变量函数的系统和方法

本发明一般涉及面向对象计算机系统，其中，一解译程序以保密方式执行对象操作。本发明特别地涉及一种用于对多种方法(methods)优化调用的解译程序的改进，所述方法的本来目的是对专用变量进行访问、修改所述专用变量、或回送一个恒定值。

在面向对象编程语言中，每一个对象都属于一个有时被称做对象类的特定的“类”(“class”)。所述一个对象类表示所述对象具有什么样的变量以及可以在一个对象上执行什么样的操作(“方法”)。

某些变量(即：在对象中)被标记为“专用”。这个标记表示只能使用属于与所述对象相同类的方法对所述变量进行访问或修改。它们不允许由其它的类进行修改或访问。它们与某些对象类的不同之处在于具有某些方法，这些方法的本来目的是访问一专用变量值、修改专用变量值、或回送一个恒定值。通过建立这种方法，该类的实现者(implementor)可以更好的隐藏实现类的详情。它还可以给予所述实现者(implementor)一个较大的自由度，再次实现该类，而不需要该类的所有用户去重新编译他们的代码。

但是，比起变量访问来讲，方法调用通常是非常昂贵的(即：占用太多的CPU 时间)。类似的，方法调用与对恒定值进行的访问相比也是非常昂贵的。

在适当的时候，某些优化编译程序自动地将一个方法调用转换成一个简单的变量访问或修改，这有时被称之为“直接插入”(in - lining)。但是，这种方案在保密环境内由于下述两个原因而不能接受：

1)在优化代码结果中，将出现所述优化后的代码直接使用其它类的一个对象的专用变量的情况。但是，一个保密运行系统将注意到这一点并标记一个保密违反。特别是，一个保密运行系统通常决不允许一方法对另一类对象内部的专用变量进行访问；和

2)如果可能不存在任何一个人都具有针对该对象类的所述“老”的规定的优化编译代码(即：利用访问专用变量的方法的老版本)的话，那么，该原始类的程序设计者将丧失修改该执行的能力。

本发明的一个目的就是要在不建立一个持久的修正程序的前提下优化运行时调用解译程序的方法，所述其方法的本来目的是要对所述专用变量或恒定值进行访问。

本发明的另一个目的是用于避免使一个类的方法对另一类对象的专用变量进行访问,在不禁止所述解译程序的一般保密措施的情况下，优化一个运行期间的解译程序，以有效的执行多种方法，所述方法的本来目的是以避免保密违反的方式对专用变量或恒定值进行访问。

总之，本发明是一个程序解译程序，该解译程序用于解译在一个具有一存储器的计算机系统中的面向对象程序，所述存储器用于存储多种类的多个对象和多个过程。在一个最佳实施例中，保密程序解译程序执行一个特殊的检查，首先，它执行一个方法调用，以确定所调用的方法的本来目的是否去访问所调用方法的类的一个实例的专用变量值、去修改所调用方法类的一个实例的专用变量值、或回送一个恒定值。当是这种情况时，修改正在执行的所述方法的解译程序的内部表达，以使其直接对所调用方法类的一个事例的专用变量进行访问或直接对所存储的调用方法的恒定值进行访问。

由所述程序解译程序内部存储的修改后的方法表达使用在一般源代码程序中得不到的多个专门的“特许”装载和存储指令，它们允许对位于正在执行方法以外的其它类的对象事例中的专用变量以及恒定值进行访问。当执行所述方法的修改部分时，使用特许装载或存储的指令由所执行的方法直接对专用变量或恒定值进行访问，这种使用避免了由所述程序解译程序标记一个保密违反。

另外，当整个程序执行完成时，所述方法修改后的内部表达被清洗(flush)。其结果是该被执行方法的修改是短暂的。如果在程序应用之间修改所述多个调用方法中的任意一个所说程序，正在执行调用方法诸如去修正指定给专用变量的值或恒定值或者具有不再对专用变量进行简单访问而是对一个值进行计算的方法，那么，在这种后续的执行程序当中将使用所述调用方法的修正版本，借此以保持对相关对象类进行修改的能力。

通过结合附图的详细描述和所附的权利要求，本发明的其它目的、特性将变得更清楚。

图 1 是结合本发明一个最佳实施例的计算机系统的方框图;

图 2 是在本发明一个最佳实施例中用于一个对象的数据结构的方框图;

图 3 是用于具有多个简单方法的对象类的数据结构的方框图;

图 4 概念性地示出了本发明装载和优化处理的方法;

图 5 是本发明一个最佳实施例中使用的程序解译程序过程的流程图。

参看图 1，其中示出了一个具有多客户(client)计算机 102 和多服务器计算机 104 的分布式计算机系统 100。在最佳实施例中，每一个客户计算机 102 都通过互联网络 103 被连接到所述服务器 104 上，当然还可以使用其它的通信连接类型。在大多数客户计算机都是诸如 Sun 工作站、IBM 兼容计算机和 Macintosh 计算机等的台式(desktop)计算机的同时，实际上，各种类型的计算机都可以被用做客户计算机。在最佳实施例中，每个客户计算机都包括一个 CPU 105、一个通讯接口 106、一个用户接口 107、一个存储器 108。所述的存储器 108 存储:

· 一个操作系统 109;

· 一个互连网络通信管理程序 110;

· 一个字节代码程序检验程序 112，该检验程序用于检验一个规定的程序是否满足某个预先规定的完整性准则(integrity criteria);

· 一个字节代码程序解译程序 114，用于执行一个应用程序;

· 一个类装载程序 116，该类装载程序将对象类装载到用户的地址空间内，并利用所述字节代码检验程序去检验与每个装载的对象类相关的所述多个方法的完整性;

· 至少一个类库(repository)120，用于局部(locally)地存储能够使所述计算机 102 的用户使用和/或得到的对象类 122;

· 至少一个对象库(repository)124，用于存储对象 124，所述对象 124 是存储于对象库 120 中所述对象类的多个对象的事例。

在最佳实施例中，所述操作系统 109 是一个面向对象的多任务操作系统，该系统支持在每个规定的地址空间内多线索执行。但是，本发明可被用于其它类型的计算机系统中，这些计算机系统包括不具有操作系统的计算机系统。

所述类装载程序 116 通常是在一个用户首先启动一个执行过程时被调用的，该执行程序要求产生一个适当对象类的对象。所述类装载程序 116 装入适当的对象类并调用字节代码程序检验程序 112 去检验装载对象类中所有字节代码程序的完整性。如果所有的方法都被成功的加以检验，那么，产生一

个对象类的对象事例，并且所述字节代码解译程序 114 被调用以执行用户要求的过程，这通常被称做是一个方法。如果用户所要求的过程不是一个字节代码程序或者允许执行该非字节代码程序(这不在本文件的范围之内)，那么，所述程序由一个编译程序的执行者(未示出)加以执行。

无论在什么时候，当一个执行的字节代码程序遇到对还没有被装入到所述用户地址空间的对象类的对象方法的调用时，就要调用所述类装载器。所述类装载器 116 就要再次装载适当的对象类，并调用字节代码程序检验程序 112 去检验在装载的对象类中所有字节代码程序的完整性。在很多情况下，所述对象类是从诸如图 1 所示一个服务器 104 的远程定位计算机(remotely located CPU)进行装载的。如果在所装载对象类中的所有方法都被进行了成功的检验，那么，就要产生所述对象类的一个对象事例，并且调用字节代码解译程序 114 去执行所调用的对象方法。

如图 1 所示，所述字节代码解译程序 114 包括工作阵列 130，所有当前装载方法的工作表达都被暂时存储在该工作阵列 130 中。所述工作表达被内部存储到所述解译程序中，并且可以被动态地进行修改以优化执行速度，这将在下面详细地加以描述。

在最佳实施例中，所述字节代码程序解译程序 114 还包括一个保密过程 132 或多个指令，用于避免一定数量的相反于保密程序执行请求的程序实践，所述的保密程序或多个指令包括多个保密指令其用于避免在一种方法中的标准装载和存储指令直接访问在其它类事例的一个对象中的专用变量。当试图利用所述程序解译程序执行任意一个这种指令时，它将把这个指令标记为一个保密违反并中止执行含有这个指令的方法。

所述字节代码程序解译程序 114 还包括一个函数调用置换过程 134，用于将过程调用置换到具有一些特殊指令的某个简单方法类型上，所述特殊指令用于对相关的专用变量直接进行访问或修改，或用于直接装载一个相关的恒定值。

用于对象的数据结构

图 2 示出了在本发明的最佳实施例中用于一个对象的数据结构 200。对象类 A 的一个对象具有对象句柄(object handle)202，该对象句柄 202 包括到与所述对象相关的方法的一个指针 204 和到与所述对象相关的数据阵列 208 的一个指针 206。

到所述对象方法的指针204实际上是一个到相关对象类方法的非直接指针。特别是，所述方法指针 204 指向一个用于对象的对象类的虚拟函数表(VFT)210。每个对象类都具有一个 VFT 210，该 VFT 210 包括所述对象类方法 214 中每一个的指针 212。所述 VFT 210 还包括用于所述对象类并被称之为类描述符 218 的一个数据结构的指针 216。除了与这里不相关的项(item)以外，所述类描述符 218 包括用于由所述对象类方法使用的多个变量中每一个变量的数据阵列区距(offset)220(表示在数据阵列 208 中的什么地方存储所述变量值)。另外，对于每个数据区距(offset)项 220 来讲，所述的类描述符包括一个变量标识(即：所述变量的名字)加上一个表示所述变量数据类型(例如整数)的指示符和一个表示所述变量是否是一个专用变量的指示符。在某些实施例中，对象的结构比起图 2 所示的更加复杂，但这些新增加的结构元件与在该文本中所讨论的内容不相关。

图 3 示出了一个用于存储具有若干“简单方法”的一个对象类方法 230 的数据结构 122 - A。就此文件而言，术语“简单方法”的含义被规定为是一种方法，它的特有(sole)功能是(A)回送一个专用变量值，其中，所述专用变量是所述简单过程专用的，(B)把一个规定值存储到所述专用变量中，或(C)回送一个恒定值。

所述字节代码程序解译程序的保密过程132避免使一个类的任一方法对另一类对象的专用变量进行直接访问。

参看图 4，与在一个对象类中的方法相关的程序码一开始就被复制到(copied into)所述解译程序的工作阵列中，以形成一个装载方法的工作内部表达。然后可以利用所述解译程序以各种方法对该方法的那个最初工作表达进行修改，以产生所述方法工作表达的优化形式。在本发明的情况下，所述方法的工作表达被进行修改，以使得对简单方法的过程调用在计算方面更加有效。

优化方法解释方法论(Interpretation Methodology)

表 1 包含有与本发明相关的部分程序解译程序过程的伪代码表达。在表 1 中使用的伪代码基本上是一个使用通用计算机语言协定的计算机语言。在这里所使用的伪代码被单独开发出来以用于本描述的目的，它的设计很容易被本技术领域的任意一个计算机程序设计者所理解。

参看图 5 和示于表 1 中所述程序解译程序过程的伪代码，当需要执行一

个方法时，所述方法的一个工作拷贝被装入到解译程序的工作阵列(260)中。在由解译程序执行所述方法期间，该解译程序选择下一个要被执行的指令(262)。如果所选择的指令是一个正在第一次被执行的方法(264 - Y)并且所调用的方法是一个简单的方法，该简单方法的特有(sole)功能是(A)回送一个专用变量值，其中，所述专用变量是该简单过程所专用的，(B)把一个规定的值存储到所述专用变量中，或(C)回送一个恒定值(266 - Y)，那么，利用相应的直接访问指令替换所述的方法调用(268)。

在最佳实施例中，利用将一个基准专用变量的值推入(pushes)所述解译程序的操作栈的特殊目的装载指令置换其特有功能是回送一个专用变量值的简单方法的方法调用:

GetVarSPC 专用变量

其中，“ GetVarSPC ”是 Get Variable(取得可变)指令的一种特殊形式，该指令被从禁止一种方法对另一种方法的专用变量进行直接访问的一般保密限制中去除(exempted)了。

在最佳实施例中，利用将一个值从解译程序操作栈存储到基准专用变量中的特殊目的存储指令置换其特有功能是将一个规定值存储到一个规定专用变量中的简单方法的方法调用:

SetVarSPC 专用变量

其中，“ SetVarSPC ”是栈到变量存储指令的一种特殊形式，该指令被从禁止一种方法对另一类对象的专用变量进行直接访问的一般保密限制中去除(exempted)了。

在最佳实施例中，利用取得一个恒定值的指令置换其特有功能是回送一个恒定值的简单方法的方法调用:

Get ConstantValue (取得恒定值)

其中，“ Get ”是用于将一个规定值推入所述解译程序操作栈的指令。

在正被执行方法的工作表达被刷新以后，不管是在步骤 264 、 266 还是步骤 268 ，所述解译过程的保密程序都要确定执行所选择的下一指令是否会违反任意一个保密限制(270)。如果不违反，那么，就执行所选择的指令(272)。如果执行所选择的指令将违反任意一个保密限制，诸如是在访问专用变量方面的限制，那么，将标记一个保密违反并中止执行所述的方法(274)。

总之，本发明通过利用等效直接插入直接访问指令置换某些方法调用来

优化某些种类的简单方法调用的执行，并以如下的方式来执行优化：每当所述调用方法被重新装载并加以执行时，就要重新产生多个直接插入指令，借此以保证由所述程序所有者或出版者所研制的调用简单方法的任意一个版本在所述调用方法的后续执行过程中被反映出来。

上述参照少数几个特定实施例对本发明的描述，仅仅是对本发明作出的示意性的说明，并且不能被认为是对本发明的限制。在不脱离由所附权利要求规定的本发明的精神和范围的情况下，本专业技术领域内的普通技术人员可以作出多种修改。

表 1

程序解译程序的伪代码表达

```
过程: 解释程序(方法)
{
将方法装入到内部工作阵列中
Do Forever
    {
    Case(将要执行的下一程序的状态):
        {
        Case=除 CetVarSPC, SetVarSPC 或 Method call 以外的任意其它的指令
            {
            标准处理，与本发明不相关
            }
        Case = GetVarSPC 或 SetVarSPC
            {
            在除正被执行方法以外的方法中中止针对访问专用变量的一般保
            密限制的同时，执行向堆栈装载或存储来自堆栈的指令。
            }
        Case = 方法调用
            {
```

如果这是此方法调用第一次被执行，由于所调用的方法已被装入
{
如果所调用的方法的功能仅仅是读出一个专用变量，那么，对于所调用的方法去读出那个专用变量来讲，它将不是一个保密违反
{
在调用方法的内部表示中用一特殊指令替换方法调用，所述特殊指令直接访问所述专用变量并将它的值装入到所述操作栈中:
GetVarSPC 专用变量
}
如果所述调用方法的功能仅仅是把一个值存储到一个专用变量中，那么对于所调用方法把一个值存储到那个专用变量中来讲，它将不是一个保密违反
{
在调用方法的内部表示中用一特殊指令替换方法调用，所述特殊指令直接对所述专用变量进行访问并将一个来自操作栈的值存储到那个专用变量中:
SetVarSPC 专用变量
}
如果所调用方法的功能仅仅是回送一个恒定值
{
在调用方法的内部表示中用一特殊指令替换方法调用，所述特殊指令直接将所述恒定值装入到所述操作栈中:
Load ConstantValue
}
当情况可以应用标准保密限制时，执行结果指令，或未变化指令。
}/*end of Case = Method Call section/*
}/*end of Case Statement*/
}/*end of Do Forever loop*/

/*Execution of Method has completed*/

从所述解译程序中清洗方法的工作表示

返回

}

5

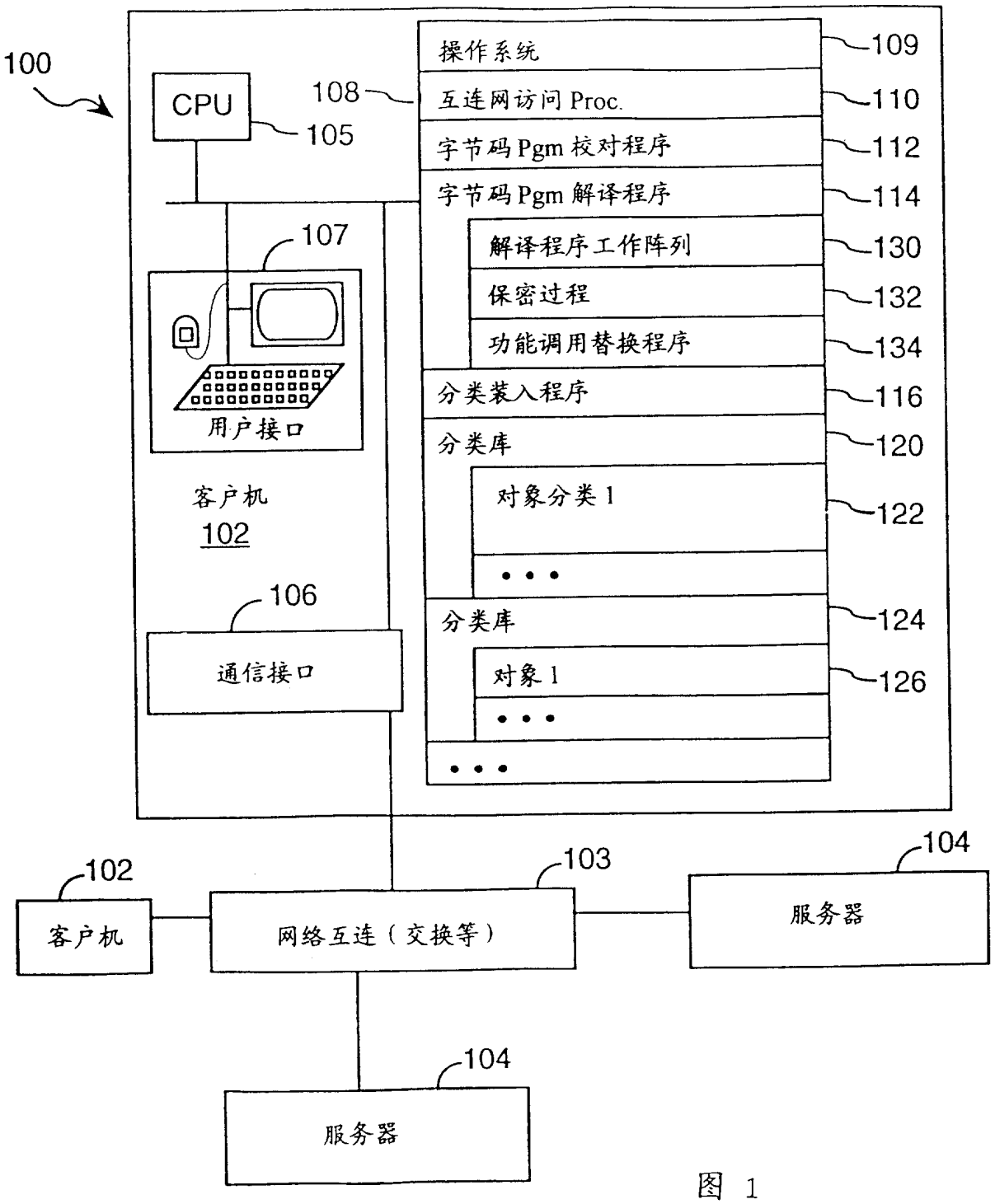

100
CPU
108
105
操作系统
109
互连网访问 Proc.
110
字节码 Pgm 校对程序
112
字节码 Pgm 解译程序
114
解译程序工作阵列
130
保密过程
132
功能调用替换程序
134
107
用户接口
客户机
102
分类装入程序
116
分类库
120
对象分类 1
122
106
通信接口
分类库
124
对象 1
126
102
客户机
103
网络互连（交换等）
104
服务器
104
服务器

图 1

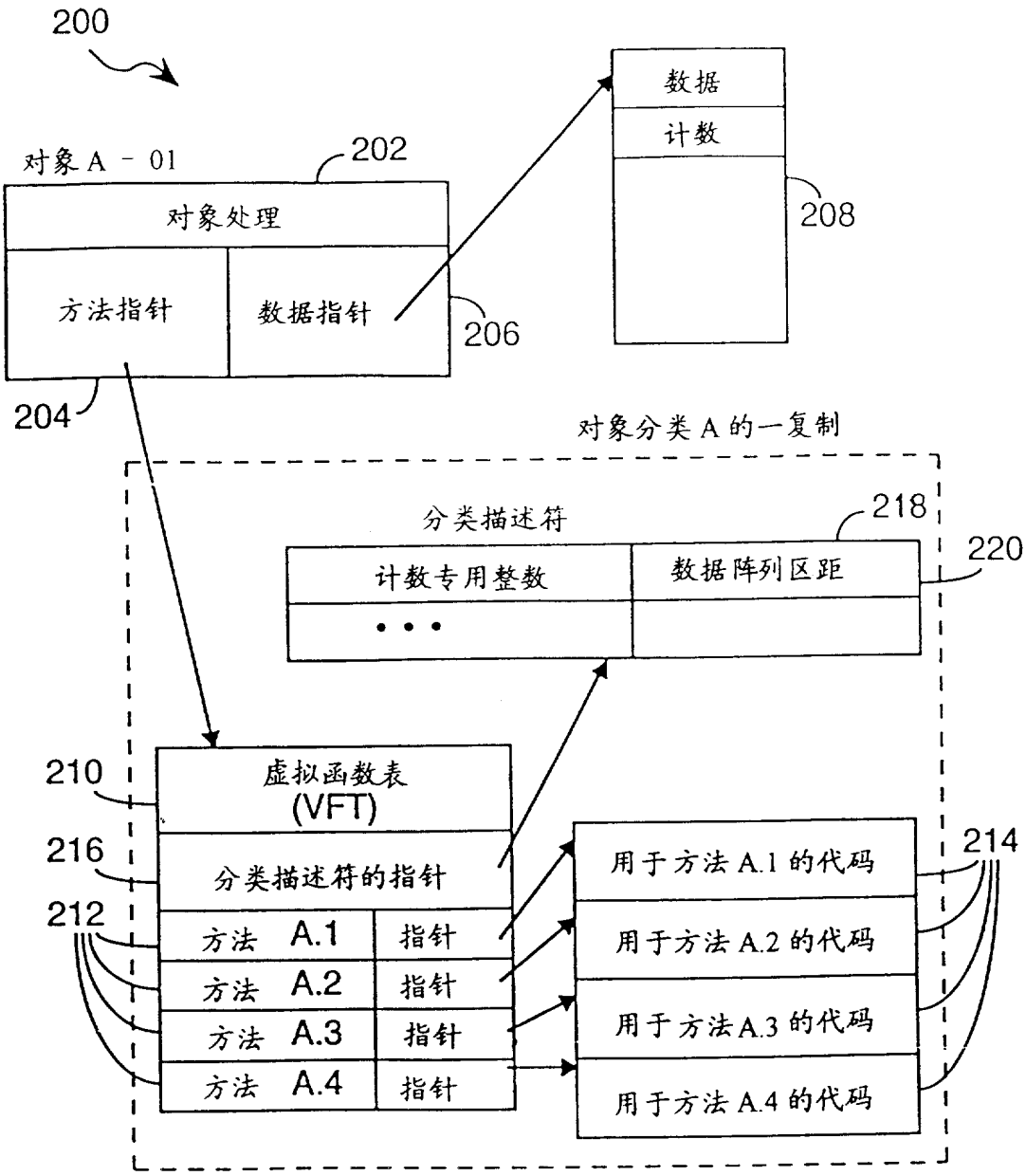

200
对象 A - 01
202
对象处理
方法指针
数据指针
206
204
数据
计数
208
对象分类 A 的一复制
分类描述符
218
220
计数专用整数
数据阵列区距
• • •
210
虚拟函数表
(VFT)
216
分类描述符的指针
212
方法 A.1
指针
方法 A.2
指针
方法 A.3
指针
方法 A.4
指针
214
用于方法 A.1 的代码
用于方法 A.2 的代码
用于方法 A.3 的代码
用于方法 A.4 的代码

图 2

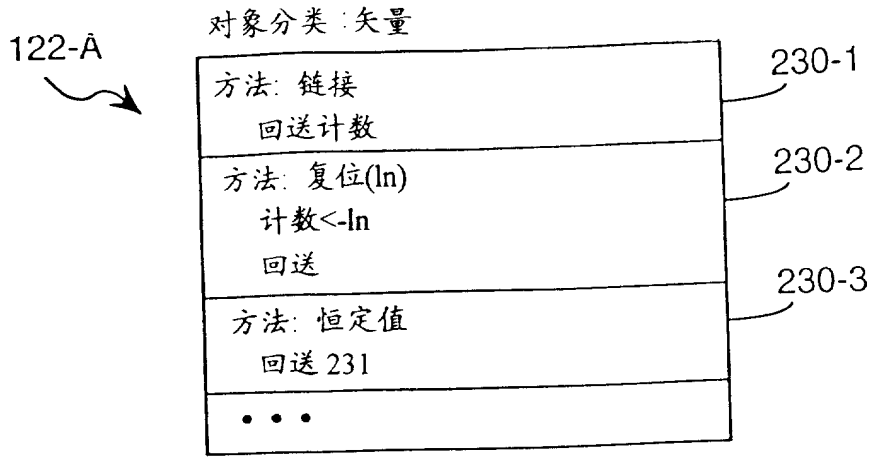

122-A
对象分类 :矢量
方法: 链接
回送计数
230-1
方法: 复位(ln)
计数<-ln
回送
230-2
方法: 恒定值
回送 231
230-3
• • •

图 3

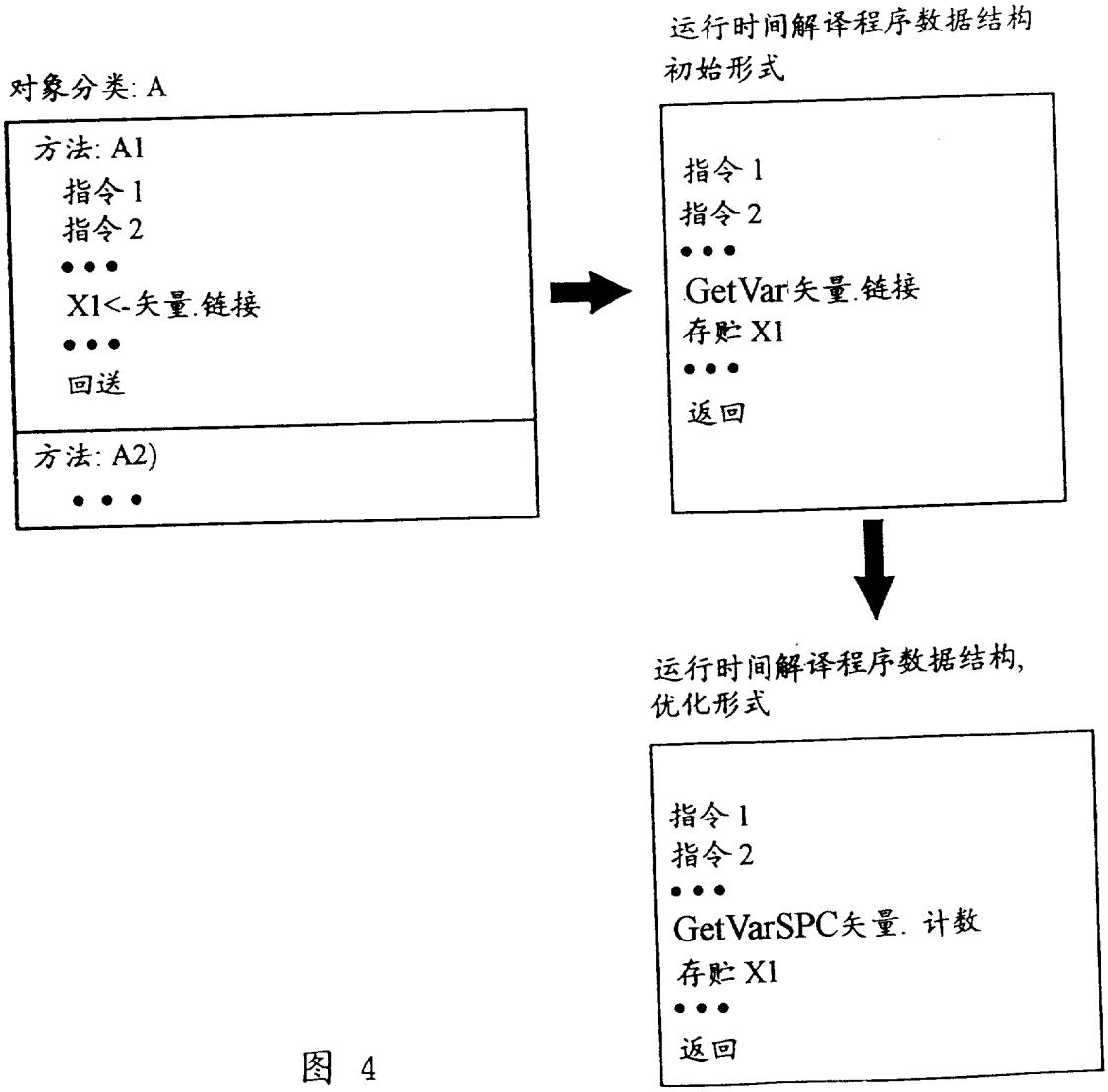

对象分类: A
方法: A1
指令 1
指令 2
• • •
X1<-矢量.链接
• • •
回送
方法: A2)
• • •
运行时间解译程序数据结构
初始形式
指令 1
指令 2
• • •
GetVar 矢量.链接
存贮 X1
• • •
返回
运行时间解译程序数据结构,
优化形式
指令 1
指令 2
• • •
GetVarSPC矢量. 计数
存贮 X1
• • •
返回

图 4

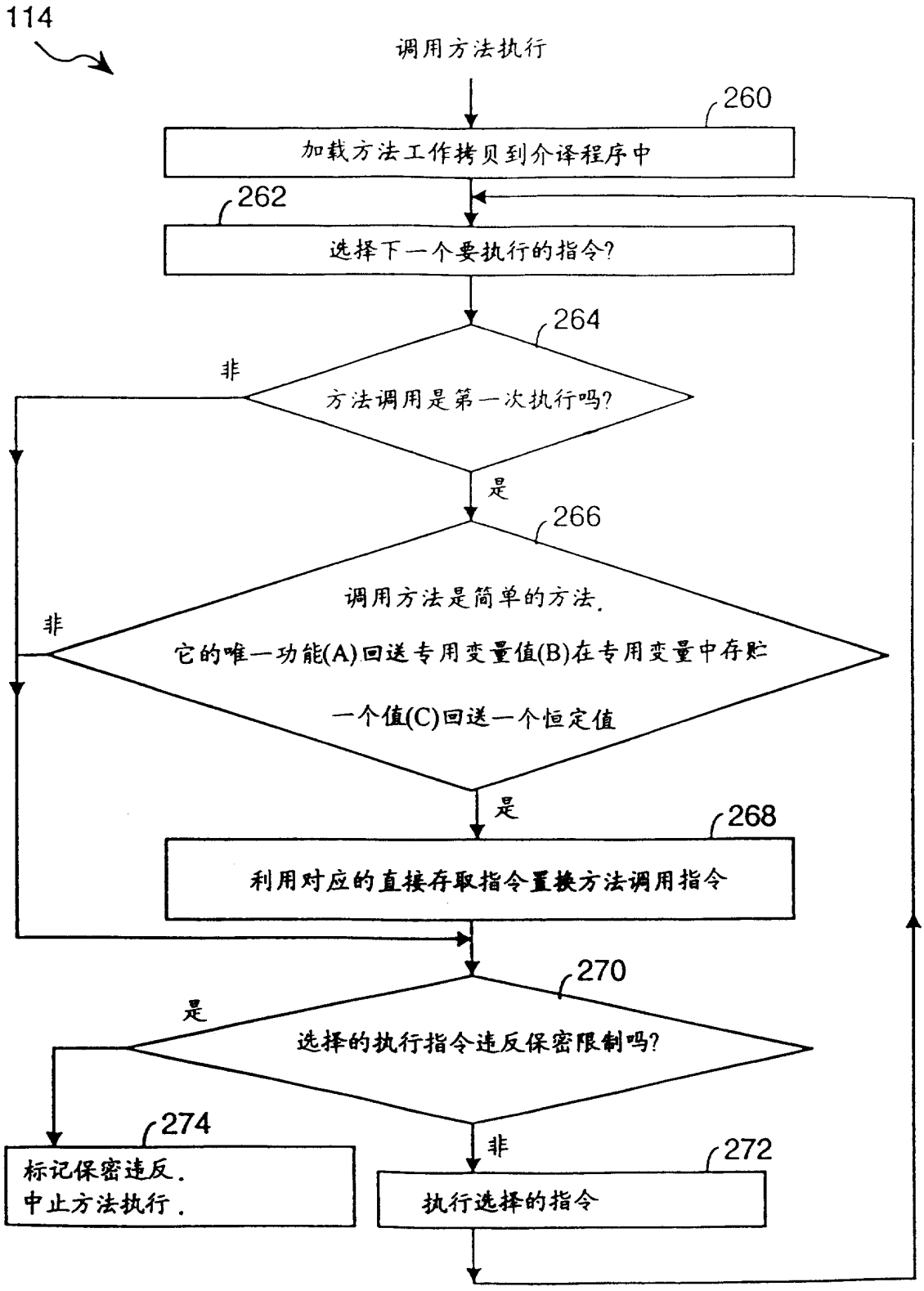

114
调用方法执行
260
加载方法工作拷贝到介译程序中
262
选择下一个要执行的指令?
264
非
方法调用是第一次执行吗?
是
266
调用方法是简单的方法.
非
它的唯一功能(A)回送专用变量值(B)在专用变量中存贮
一个值(C)回送一个恒定值
是
268
利用对应的直接存取指令置换方法调用指令
270
是
选择的执行指令违反保密限制吗?
274
标记保密违反.
中止方法执行.
非
272
执行选择的指令

图 5