

(12) 发明专利申请

(10) 申请公布号 CN 102541731 A

(43) 申请公布日 2012. 07. 04

(21) 申请号 201110430699. 1

(22) 申请日 2011. 12. 08

(30) 优先权数据

12/963, 630 2010. 12. 09 US

(71) 申请人 微软公司

地址 美国华盛顿州

(72) 发明人 M·萨菲 L·杜宾斯基

(74) 专利代理机构 上海专利商标事务所有限公司 31100

代理人 陈斌

(51) Int. Cl.

G06F 11/36 (2006. 01)

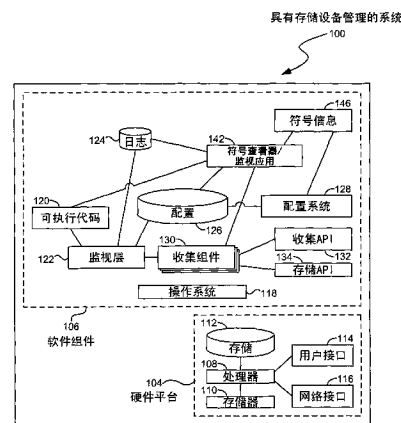
权利要求书 2 页 说明书 7 页 附图 3 页

(54) 发明名称

平台不可知诊断数据收集与显示

(57) 摘要

本发明涉及平台不可知诊断数据收集与显示。一种数据收集系统,可通过将可执行代码加载到存储器中并根据监视条件探测该代码来探测任意可执行代码并从中收集数据。所述探测可包括到在监视条件存在时可采集并存储信息的数据收集器的指针或绑定。显示模块可允许编程者浏览结果。该数据收集系统可对任何类型的本机代码和中间代码操作并可在有或没有符号表的情况下操作。



1. 一种在计算机处理器上执行的方法,所述方法包括:  
接收可执行代码;  
接收定义要在所述可执行代码中监视的条件的配置;  
根据所述条件在所述可执行代码内标识至少一个位置;  
在所述至少一个位置处探测所述可执行代码,所述探测包括对收集例程的调用;  
执行所述可执行代码,直到所述条件存在;以及  
当所述条件存在时,致使所述收集例程被执行。
2. 如权利要求 1 所述的方法,其特征在于,在所述探测之前将所述可执行代码接收到随机存取存储器中。
3. 如权利要求 1 所述的方法,其特征在于,还包括:  
当所述条件存在时暂停所述可执行代码中的至少一部分;以及  
在所述收集例程已被执行后恢复所述可执行代码。
4. 如权利要求 1 所述的方法,其特征在于,还包括:  
接收所述可执行代码的符号信息;以及  
使用所述符号信息以用于所述的在所述可执行代码内标识所述至少一个位置。
5. 如权利要求 4 所述的方法,其特征在于,所述符号信息是通过所述可执行代码的反射获得的。
6. 如权利要求 1 所述的方法,其特征在于,所述可执行代码是中间代码。
7. 如权利要求 1 所述的方法,其特征在于,所述可执行代码是本机代码。
8. 如权利要求 7 所述的方法,其特征在于,所述本机代码不具有符号信息。
9. 如权利要求 1 所述的方法,其特征在于,所述收集例程收集与所述可执行代码相关联的数据。
10. 如权利要求 1 所述的方法,其特征在于,所述收集例程取回与所述可执行代码相关联的类型,串行化所述类型以创建经串行化的类型,以及在日志文件中存储所述经串行化的类型。
11. 如权利要求 1 所述的方法,其特征在于,所述收集例程收集不与所述可执行代码相关联的数据。
12. 一种系统,包括:  
处理器;  
配置系统,所述配置系统:  
读取可执行代码的符号信息;  
接收用于收集的条件;以及存储所述条件;  
运行时执行器,所述运行时执行器:  
接收所述可执行代码;  
根据所述条件监视所述可执行代码;  
执行所述可执行代码;以及  
标识用于收集的所述条件并启动收集例程;  
收集器,所述收集器:  
执行所述收集器例程;

- 寻找所述条件中定义的数据 ;以及  
存储所述数据。
13. 如权利要求 12 所述的系统,其特征在于,还包括:  
显示所述数据的日志查看器。
14. 如权利要求 13 所述的系统,其特征在于,当显示所述数据时使用所述符号信息。
15. 如权利要求 14 所述的系统,其特征在于,所述符号信息是从所述可执行代码的反射导出的。
16. 如权利要求 14 所述的系统,其特征在于,所述可执行代码是中间代码。
17. 一种系统,包括:  
处理器;  
配置系统,所述配置系统:  
接收用于收集的条件,所述条件包括可执行代码中的方法调用 ;以及  
存储所述条件;  
运行时执行器,所述运行时执行器:  
接收所述可执行代码;  
将所述可执行代码加载到所述处理器能够访问的存储器中以执行所述可执行代码;  
通过在所述可执行代码中添加指针来根据所述条件探测所述可执行代码;  
执行所述可执行代码 ;以及  
当在所述执行期间遇到所述指针时,启动收集例程;  
收集器,所述收集器:  
执行所述收集器例程;  
寻找所述条件中定义的数据 ;以及  
存储所述数据。
18. 如权利要求 17 所述的系统,其特征在于,所述运行时执行器还:  
当遇到所述指针时暂停所述可执行代码。
19. 如权利要求 18 所述的系统,其特征在于,所述配置系统还:  
对所述可执行代码执行反射以生成符号表 ;以及  
使用所述符号表来标识所述条件。
20. 如权利要求 19 所述的系统,其特征在于,还包括:  
日志查看器,所述日志查看器:  
接收所述数据 ;以及  
使用所述符号表呈现所述数据。

## 平台不可知诊断数据收集与显示

### 技术领域

[0001] 本发明涉及平台不可知诊断数据收集与显示。

### 背景技术

[0002] 调试编程代码在最好情况下也可能是一项困难的任务。代码经常是由不同开发者写的,每个开发者具有不同的编程风格或约定集。当代码来自诸如不同供应方等不同来源时,或者当代码是在许多年间编写和增强时,问题可能扩大。

[0003] 在许多编程环境中,调试套件可允许用户以有效的方式编辑并调试代码。一般而言,这些环境可以以人类可读的形式来显示变量名、值和其他信息。然而,这可使用源代码和符号表来创建可用并高效的调试系统。这些环境对于源代码可能不可用的代码也许不会很好地操作。

### 发明内容

[0004] 一种数据收集系统可通过根据预定的监视条件监视代码来探测 (instrument) 任意可执行代码并从其收集数据。所述探测 (instrumentation) 可包括到在监视条件存在时可采集并存储信息的数据收集器的指针或绑定。显示模块可允许编程者浏览结果。数据收集系统可对任何类型的本机代码或中间代码操作并可在有或没有符号表的情况下操作。

[0005] 提供本发明内容以便以简化形式介绍在以下具体实施方式中进一步描述的一些概念。本发明内容并不旨在标识所要求保护主题的关键特征或必要特征,也不旨在用于限制所要求保护主题的范围。

### 附图说明

[0006] 在附图中:

[0007] 图 1 是示出具有用于可执行代码的数据收集系统的系统的实施例的图示。

[0008] 图 2 是示出用于准备监视可执行代码的方法的实施例的流程图。

[0009] 图 3 是示出用于捕捉可执行代码的数据的方法的实施例的流程图。

### 具体实施方式

[0010] 数据收集系统可监视任何类型的可执行代码以供调试和其他操作使用。可执行代码可被加载到计算机存储器中并且可在代码内标识各种条件。在每种条件下,用对收集例程的调用探测 (instrument) 该可执行代码。收集例程可标识要采集的各种对象,这些对象可被存储例程存储。一旦数据被收集,则虚拟化系统可查看该数据以用于调试或其他目的。

[0011] 该数据收集系统可以能够采集复杂的数据类型,这些数据类型在存储之前被串行化并且在查看之前被去串行化 (deserialize)。无论源代码和符号表是否可用,该数据收集系统可以能够收集来自任何类型的可执行代码的数据。这种能力可允许将任意可执行代码合并到调试操作中。

[0012] 贯穿本说明书,在所有附图的描述中,相同的附图标记表示相同的元素。

[0013] 当元素被称为“连接”或“耦合”时,这些元素可直接连接或耦合在一起,或者也可存在一个或多个中间元素。相反,当元素被称为“直接连接”或“直接耦合”时,不存在中间元素。

[0014] 本主题可被具体化为设备、系统、方法、和 / 或计算机程序产品。因此,本发明主题的部分或全部可以用硬件和 / 或软件(包括固件、常驻软件、微码、状态机、门阵列等)来具体化。此外,本发明主题可以采用其上嵌入有供指令执行系统使用或结合其使用的计算机可使用或计算机可读的程序代码的计算机可使用或计算机可读的存储介质上的计算机程序产品的形式。在本文档的上下文中,计算机可使用或计算机可读介质可以是可包含、存储、通信、传播、或传输程序以供指令执行系统、装置或设备使用或结合其使用的任何介质。

[0015] 计算机可使用或计算机可读介质可以是,例如,但不限于,电、磁、光、电磁、红外、或半导体系统、装置、设备或传播介质。作为示例而非限制,计算机可读介质可包括计算机存储介质和通信介质。

[0016] 计算机存储介质包括以用于存储诸如计算机可读指令、数据结构、程序模块或其它数据这样的信息的任意方法或技术来实现的易失性和非易失性、可移动和不可移动介质。计算机存储介质包括,但不限于,RAM、ROM、EEPROM、闪存或其他存储器技术、CD-ROM、数字多功能盘(DVD)或其他光盘存储、磁带盒、磁带、磁盘存储或其他磁性存储设备、或可用于存储所需信息且可由指令执行系统访问的任何其他介质。注意,计算机可使用或计算机可读介质可以是其上打印有程序的纸张或其他合适的介质,因为程序可经由例如对纸张或其他合适的介质的光学扫描来电子地捕获,随后如有必要被编译、解释,或以其他合适的方式处理,并且随后存储在计算机存储器中。

[0017] 通信介质通常以诸如载波或其他传输机制之类的已调制数据信号来具体化计算机可读指令、数据结构、程序模块或其他数据,并且包括任何信息传送介质。术语“已调制数据信号”可被定义为其一个或多个特性以对信号中的信息编码的方式设置或改变的信号。作为示例而非限制,通信介质包括诸如有线网络或直接线连接之类的有线介质,以及诸如声学、RF、红外及其他无线介质之类的无线介质。上述的任何组合也应包含在计算机可读介质的范围内。

[0018] 当本主题在计算机可执行指令的一般上下文中具体化时,该实施例可包括由一个或多个系统、计算机、或其他设备执行的程序模块。一般而言,程序模块包括执行特定任务或实现特定抽象数据类型的例程、程序、对象、组件、数据结构等。通常,程序模块的功能可在各个实施例中按需进行组合或分布。

[0019] 图 1 是实施例 100 的示意图,示出可包括用于可执行代码的数据收集系统的系统的实施例。实施例 100 是可执行任意可执行代码并监视该可执行代码内的某些条件的硬件和软件平台的简化示例。

[0020] 图 1 的示意图示出了系统的各个功能组件。在一些情况下,组件可以是硬件组件、软件组件、或者硬件和软件的组合。一些组件可以是应用级软件,而其他组件可以是操作系统级组件。在一些情况下,一个组件到另一组件的连接可以是紧密连接,其中两个或更多个组件在单个硬件平台上操作。在其他情况下,连接可通过跨长距离的网络连接来进行。各实施例可使用不同的硬件、软件、以及互连体系结构来实现所描述的功能。

[0021] 实施例 100 示出可探测 (instrument) 任意可执行代码以用于调试目的的计算机系统。可定义可用于标识可执行代码内可调用收集例程的地方的各种条件。收集例程可在可执行代码内定位各种数据类型并致使该数据类型被存储。

[0022] 该数据收集系统可在设备 102 上工作。设备 102 被示为具有硬件组件 104 和软件组件 106。图示的控制器设备 102 表示常规计算设备,但是其他实施例可具有不同配置、架构、或组件。

[0023] 在许多实施例中,控制器设备 102 可以是个人计算机或代码开发工作站。控制器设备 102 还可以是服务器计算机、台式计算机、或类似设备。在一些实施例中,控制器设备 102 仍然还可以是膝上型计算机、上网本计算机、平板 (tablet 或 slate) 计算机、无线手持设备、蜂窝电话、或任何其他类型的计算设备。

[0024] 硬件组件 104 可包括处理器 108、随机存取存储器 110、以及非易失性存储 112。硬件组件 104 还可包括用户接口 114 和网络接口 116。在一些实施例中,处理器 108 可由若干处理器或处理器核组成。随机存取存储器 110 可以是处理器 108 容易访问和寻址的存储器。非易失性存储 112 可以是在关闭设备 102 之后持久保持的存储。非易失性存储 112 可以是任何类型的存储设备,包括硬盘、固态存储器设备、磁带盒、光学存储、或其他类型的存储。非易失性存储 112 可以是只读,或能够读 / 写。

[0025] 用户接口 114 可以是能够显示输出并且从用户接收输入的任何类型的硬件。在许多情况下,输出显示可以是图形显示监视器,但是输出设备可包括灯光和其他视觉输出、音频输出、动力致动器输出、以及其他输出设备。常规输入设备可包括键盘和定点设备,诸如鼠标、指示笔、跟踪球、或其他定点设备。其他输入设备可包括各种传感器,包括生物测定输入设备、音频和视频输入设备、以及其他传感器。

[0026] 网络接口 116 可以是到另一计算机的任何类型的连接。在许多实施例中,网络接口 116 可以是有线以太网连接。其他实施例可包括基于各种通信协议的有线或无线连接。

[0027] 软件组件 106 可包括操作系统 118,各种应用和服务可在操作系统 118 上操作。操作系统可在执行例程和执行硬件组件 104 之间提供抽象层,并且可包括各种例程和与各个硬件组件直接通信的功能。

[0028] 实施例 100 示出了可用于基于可执行代码中的若干条件来收集和存储数据的软件架构。在示例架构中,一些可执行代码 120 可关于监视层 122 执行。监视层 122 可填充日志 124,该日志可使用日志查看器 136 来读取。监视层 122 可与各种收集组件 130 交互,这些收集组件 130 可定义在执行过程中满足某些条件时要收集的数据。

[0029] 收集组件 130 可具有收集 API 132 和存储 API 134。收集 API 132 可包括可采集数据并可从收集组件 130 调用的各种预定义例程。收集 API 132 可包括从不同的来源采集数据的功能。

[0030] 例如,一种编程环境或语言可使用一种格式存储一数据类型而另一种编程环境或语言可以用不同格式存储相同的数据类型。收集 API 132 可具有收集例程,收集例程可以用数据被存储的格式来采集数据,然后存储 API 134 可将该数据转换为公共格式以供存储和取回。

[0031] 存储 API 134 可在存储之前将复杂的数据类型串行化。在许多实施例中,存储 API 134 可存储该数据和可用于填充或重构该数据类型以供查看或其他目的的元数据。

[0032] 在许多实施例中,收集组件 130 可以是用于定义当遇到一条件时可收集什么的个体例程、语句、或其他机制。收集组件 130 可由可对可执行代码 120 执行调试的开发者编写。

[0033] 该数据收集系统可为开发者提供许多工具以探测 (instrument) 经编译的可执行代码并在该可执行代码执行时监视、收集并查看对象。为了监视任何类型的可执行代码,开发者可定义数据收集可发生的条件,然后定义要收集的数据。该数据收集系统可针对该条件监视该可执行代码,然后从收集 API 132 调用适当的例程来收集所标识的数据,然后从存储 API 134 调用存储例程。

[0034] 该数据收集系统可允许将广泛的探测 (instrumentation) 和监视添加到以前编译的可执行程序 (executable)。当用于创建该可执行程序的源代码不可用时,该数据收集系统可以是有用的。因为探测 (instrumentation) 和监视可在编译之后进行,所以可执行代码在监视之前可不被改变。

[0035] 在一些实施例中,数据收集系统可以能够监视来自不同来源的代码。例如,较老的传统应用编程接口可供新开发的应用操作。传统应用编程接口 (API) 可能来自制造商而开发者可能不具有任何源代码。新开发的应用可以用更新近的编程语言编写,而该 API 和该应用两者都可在执行过程中交互。开发者可使用该数据收集系统来收集、显示并审查调试信息以探测 (instrument) 该 API 和应用两者。

[0036] 收集组件 130 可定义当遇到一所监视的条件时要从多个源收集的数据。在上面的示例中,收集组件可定义要从该 API 以及从该应用收集的信息。因为所收集的数据在该 API 和应用内可能是不同格式的,所以收集 API 132 可将这两种数据格式转换为公共数据格式以供存储 API 134 使用。

[0037] 配置 126 可定义何时收集数据。配置 126 可包含定义可收集数据的条件的表达式,以及在满足该条件时可被调用的特定的收集组件 130。配置系统 128 可以是用于创建数据收集的各种条件的交互式应用。

[0038] 在一些实施例中,在配置 126 中定义的条件可以是复杂的表达式,该表达式可包括多个输入、通配符、代数表达式或其他组件。条件可包括在汇编件或动态链接库级、类型或接口级、或方法级的引用。使用各种表达式和条件级,开发者可以能够创建很宽范围的可执行数据收集的条件。

[0039] 在汇编件或动态链接库级,条件可包括:对具有公钥权标的某些名称的汇编件或动态链接库、库的版本、与库相关联的文化、库的各种属性、库内的模块的名称、以及其他参数的各种引用。在一简单示例中,一条件可被定义为只要可能访问以字母 'A' 开头的汇编件时就捕捉数据。另一示例可以是定义一条件以在特定版本和文化的特定库被访问时捕捉数据。

[0040] 在类型或接口级,条件可基于类型或接口名称、有关该类型或接口的各种元数据、由类型实现的任何接口、可见性参数、接口的类型的属性、到特定汇编件的指针、或其他参数来定义。

[0041] 在方法级,条件可基于方法名称、关于方法的元数据、传递到方法的自变量、从方法的返回值、显式接口实现、方法的各种属性、可见性设定或其他参数来定义。

[0042] 可执行代码 120 可具有或可不具有分开的符号信息。在一些编程环境中,可执行代码 120 可具有可被即时 (just-in-time) 编译器或其他运行时组件使用来将可执行代码

120 转换为可在运行时链接到各种汇编件或动态链接库的机器语言的元数据。在这些实施例中,符号信息可从可执行代码 120 直接被读取或导出。

[0043] 在符号信息可不被合并入可执行代码 120 的实施例中,符号信息可从可执行代码导出。这种导出可使用各种反射 (reflection) 技术或例程来执行,该反射技术或例程可分析可执行代码 120 并创建可包括诸如方法、类型、接口、汇编件、参数、变量或可执行代码内的其他人工件 (artifact) 等各种编程对象的名称的符号表。在一些实施例中,符号表可作为分开的文件与可执行代码 120 一起提供。

[0044] 符号信息 146 可包括用于可执行代码 120 的任何类型的符号信息。符号信息 146 内的命名和标记信息可在配置系统 128 内被使用以使用对象的人类可读且有时候有意义的标记来标识可执行代码 120 内的对象。符号信息 146 可包括可描述协助开发者在使用配置系统 128 时创建条件的数据类型的元数据。

[0045] 符号查看器和监视应用 142 可以是中央用户接口,开发者通过该中央用户接口可创建包含用于监视以及定义收集组件 130 并查看日志 124 的条件的配置 126。

[0046] 应用 142 可接收可执行代码 120 并确定可执行代码 120 中可包括的任何类型的符号信息 146。在一些情况下,应用 142 可对可执行代码 120 内的元数据执行反射或分析该元数据。在其他情况下,应用 142 可接收符号表或符号信息的其他次级来源。

[0047] 应用 142 可具有开发者可用于浏览符号信息 146 的图形用户界面或其他机制来在可执行代码 120 内定位对象以创建条件。在许多实施例中,应用 142 可允许开发者为条件创造复杂的表达式。

[0048] 然后开发者可使用应用 142 来创建各收集组件 130。开发者可通过查看符号信息 146 来定义要收集的数据。在一些实施例中,该开发者可以能够定义也可在收集组件 130 中定义的对要收集的数据的过滤器或限制。

[0049] 在致使可执行代码 120 关于监视层 122 被执行后,应用 142 可能显示来自日志 124 的所收集的数据。当符号信息 146 可用时,应用 142 可用高保真方式呈现来自日志 124 的数据。所述高保真方式包括数据类型在所述显示中的标记、格式化和组织。

[0050] 可执行代码 120 被示出为单个组件。在一些实施例中,可执行代码 120 可以是多个组件,每个组件可以用使用不同数据存储机制并具有其他不同性质的不同语言编写。应用 142 可使开发者能够创建可调用收集 API 132 来从可执行代码的各部分中的每一个部分采集复杂数据类型,将那些数据类型串行化到公共日志 124 中的收集组件 130 的单个集合。应用 142 可从日志 124 取回数据并用合并 (consolidated) 视图来呈现所述数据。

[0051] 因为该数据收集系统可使用标准化存储 API 134,所以数据可从来自不同来源的可执行代码收集并合并 (consolidate) 却用统一或合并的视图来显示。

[0052] 在一些实施例中,应用 142 的功能可被合并到图形代码开发平台中。这样的系统可具有用于创建新代码的代码编辑器和各种编译和调试功能,以及可从可与在开发的新代码交互的其他可执行代码采集数据的数据收集系统。

[0053] 图 2 是示出用于准备监视可执行程序 (executable) 的方法的实施例 200 的流程图。实施例 200 的过程是可如何创建标识数据收集事件何时可发生的条件的简化示例。

[0054] 其他实施例可使用不同的排序、附加或更少的步骤、以及不同的命名或术语来完成类似功能。在一些实施例中,各个操作或操作集合可与其他操作以同步或异步方式并行



地执行。此处所选的步骤是为了以简化的形式示出一些操作原理而选择的。

[0055] 可在框 202 中接收可执行代码,以及可在框 204 中接收符号信息。在一些实施例中,框 204 的符号信息可从该可执行代码导出,这是通过分析代码内的元数据或者通过执行反射来导出的。在一些情况下,该符号信息可以用符号表或其他数据库的形式提供。

[0056] 在框 206 中,开发者可标识要监视的条件。所述条件可以用符号信息中定义的对象来定义。例如,一方法可以被来自符号信息的标记定义。监视层可接收条件中的符号信息并将该符号条件转换成存储器位置以监视该条件。

[0057] 在框 208 中,开发者可标识当条件发生时要执行的动作。动作可被定义在如在实施例 100 的项 130 中讨论的收集组件中。与条件类似,可使用符号信息来定义动作。符号信息可被转换为标识要收集的所需数据的存储器位置或其他机器级信息。在框 208 中定义的动作可与在框 206 中定义的条件相关联。

[0058] 创建条件和动作的过程可从框 210 循环回到框 206。当在框 210 中定义了所有条件和动作时,可在框 212 中存储配置文件并且可在框 214 中存储收集组件。框 212 的配置文件可被关于可执行代码执行的监视层消耗以标识条件,然后致使该收集组件被执行。

[0059] 图 3 是示出用于捕捉可执行代码的数据的方法的实施例 300 的流程图。实施例 300 的过程是用于监视可执行代码,标识条件,然后执行收集组件以收集数据,并且存储该数据的过程的简化示例。

[0060] 其他实施例可使用不同的排序、附加或更少的步骤、以及不同的命名或术语来完成类似功能。在一些实施例中,各个操作或操作集合可与其他操作以同步或异步方式并行地执行。此处所选的步骤是为了以简化的形式示出一些操作原理而选择的。

[0061] 实施例 300 可以是可在监视正在执行的应用时执行的运行时操作的示例。实施例 300 的操作可由监视层、收集组件和各种 API 执行。

[0062] 实施例 300 的操作示出了一种其中可用对收集组件的调用探测 (instrument) 可执行代码的方法。在其他实施例中,监视层可与可执行代码并行操作以标识条件而不探测 (instrument) 该可执行代码。

[0063] 在框 302 中,可标识要执行的代码。在框 304 中可加载配置信息。在框 306 中可将可执行代码加载到存储器。

[0064] 对于框 308 中的每个条件,在框 310 中可标识该可执行代码内的该条件可能发生的点。在框 312 中可在该点处探测 (instrument) 该代码以调用相应的收集例程。

[0065] 在处理框 308 中的每个条件之后,在框 314 中可存储该可执行代码。

[0066] 在一些实施例中,在框 316 中可使用即时编译器来编译该可执行代码中的一些或一部分。

[0067] 在框 318 中该代码可开始执行。该代码可执行,直到在框 320 中所述条件中的一个可被标识为数据收集点。

[0068] 在一些实施例中,可用对收集例程的函数调用来探测 (instrument) 该可执行代码。在其他实施例中,监视层可跟踪执行过程中的变化以标识收集数据的条件。

[0069] 当在框 320 中标识了收集点时,在框 322 中可暂停该可执行代码。在可用函数调用来探测 (instrument) 该可执行代码的实施例中,该可执行代码可在该函数调用正被执行时暂停,并可在该函数调用已完成操作之后恢复。

[0070] 在一些实施例中,该可执行代码可不暂停。一个示例可以是不探测 (instrument) 该可执行代码但是可通过监视层监视该可执行代码的情况。在这样的实施例中,暂停该应用也许是可行的,也许是不可行的。

[0071] 在框 324 中可调用该收集组件并且在框 326 中可执行该收集组件。该收集组件可定义要收集什么数据,并且可在框 328 中将请求和各种参数传到数据收集 API 以采集所请求的数据。在一些实施例中,该收集组件可传递若干请求到该数据收集 API 并接收若干响应。在一些这样的实施例中,该收集组件可从具有不同数据类型定义和存储机制的不同可执行代码段收集数据。

[0072] 在框 330 中,所收集的数据可被传递到数据存储 API,该数据存储 API 可串行化并以公共格式存储该数据。在许多实施例中,该数据可从两个或更多个不同来源收集,每个来源具有不同数据存储格式,且所收集的数据可以用公共格式存储,该公共格式可对应于或可不对应于该数据来源的格式中的一种。

[0073] 在框 322 中该存储 API 可串行化该数据并且在框 334 中该存储 API 可将该数据写到日志。

[0074] 如果在框 322 中暂停该可执行代码,则在框 336 中可恢复该可执行代码且该过程可返回到框 318 以继续执行。

[0075] 对本发明的上述描述是出于图示和描述的目的而呈现的。它不旨在穷举本主题或将本主题限于所公开的精确形式,并且鉴于上述教导其他修改和变型都是可能的。选择并描述实施例来最好地解释本发明的原理及其实际应用,由此使本领域的其他技术人员能够在各种实施例和各种适于所构想的特定用途的修改中最好地利用本发明。所附权利要求书旨在被解释为包括除受现有技术所限的范围以外的其他替换实施例。

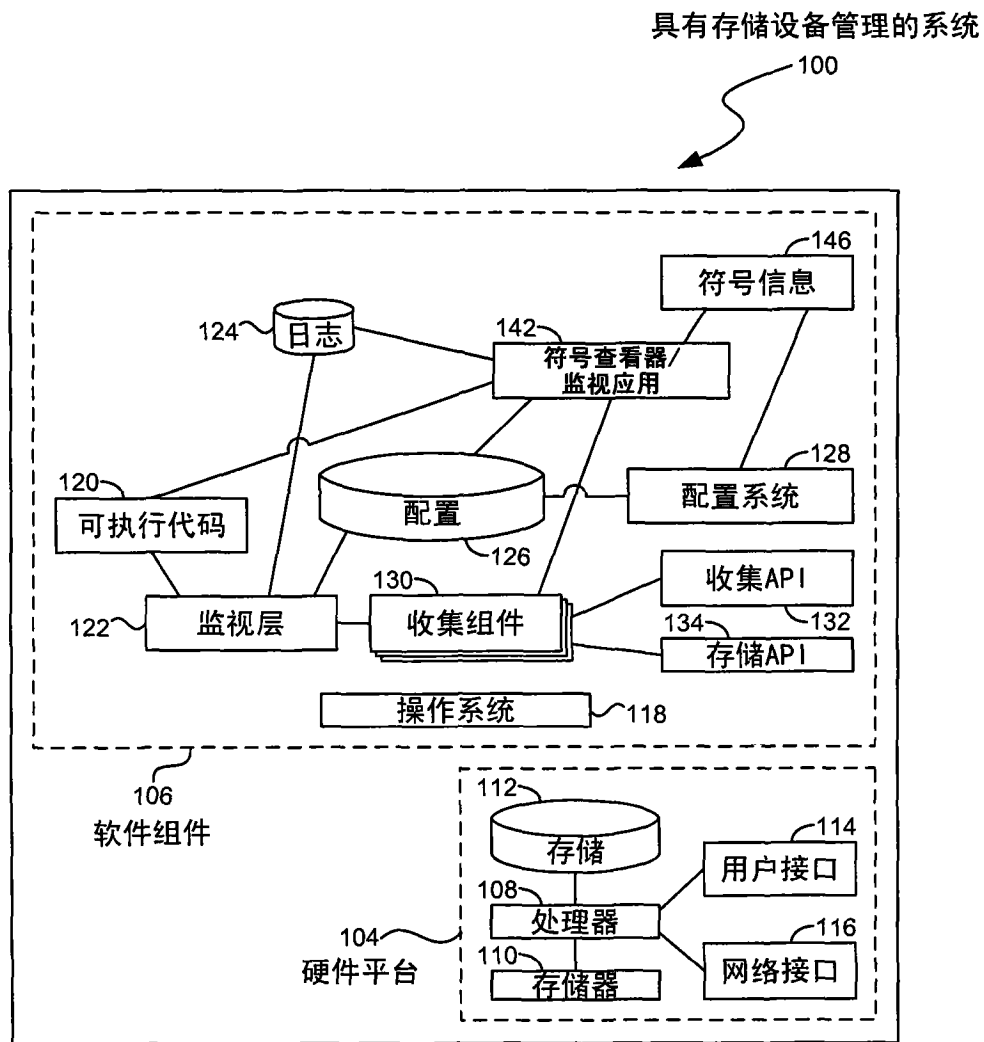


图 1

用于准备监视可执行程序的方法

200

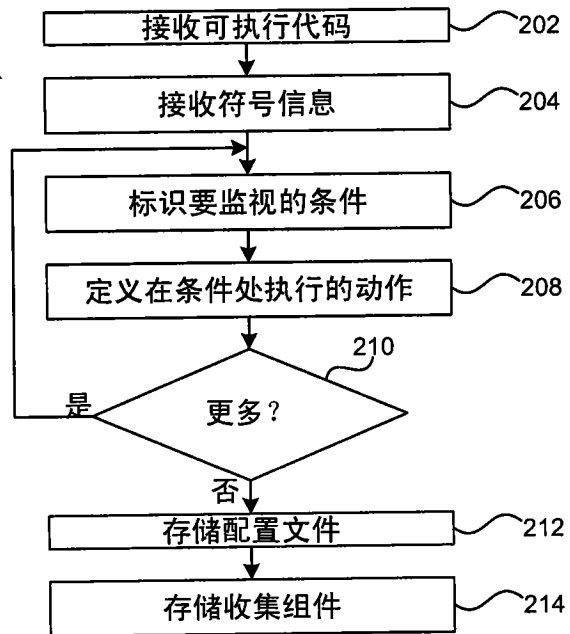


图 2

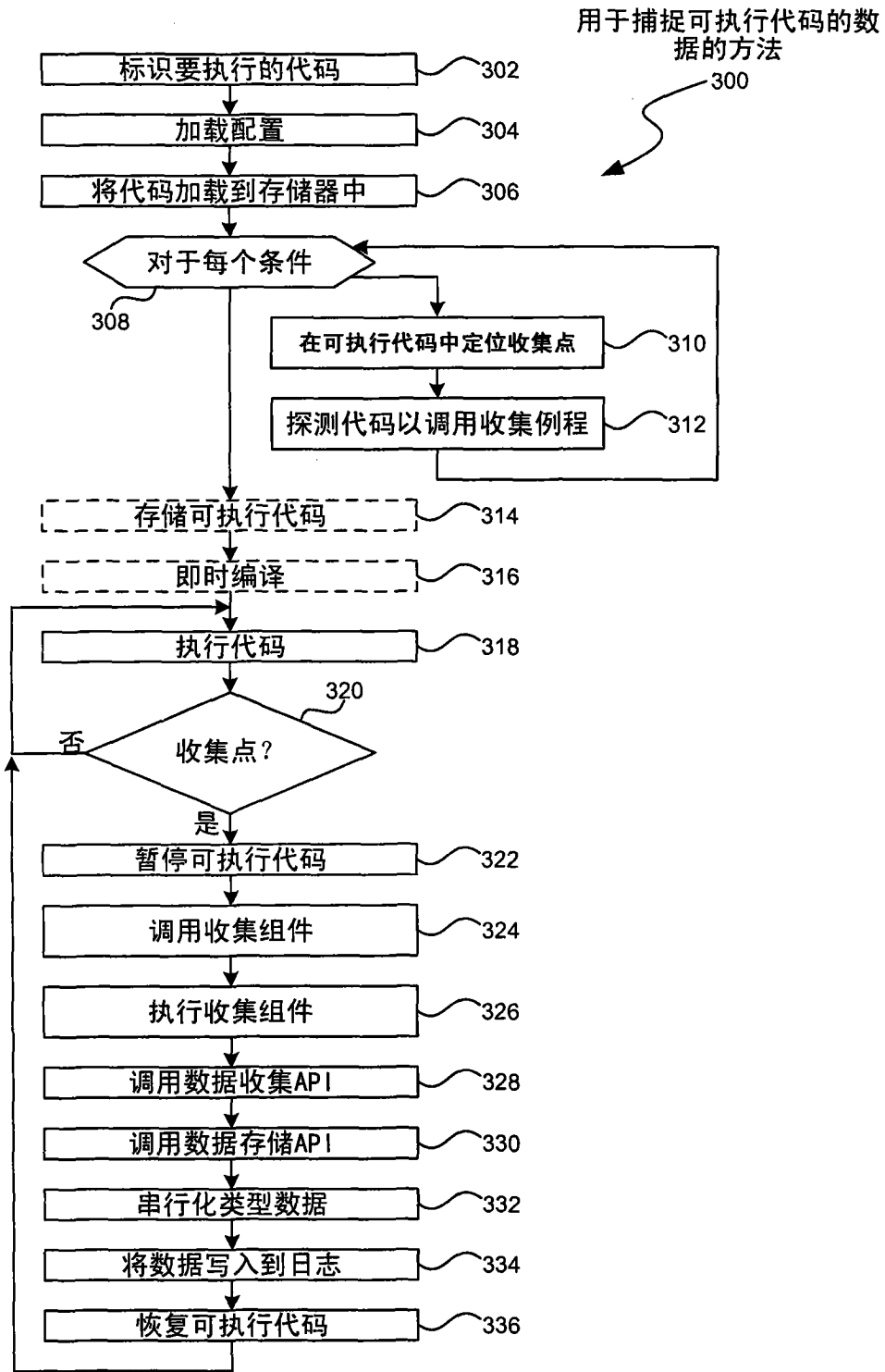


图 3