(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 105378689 A
(43)申请公布日 2016.03.02

(21)申请号 201380076247.5
(22)申请日 2013.06.14
(85)PCT国际申请进入国家阶段日
2015.10.30
(86)PCT国际申请的申请数据
PCT/US2013/045962 2013.06.14
(87)PCT国际申请的公布数据
WO2014/200511 EN 2014.12.18
(71)申请人 惠普发展公司，有限责任合伙企业
地址 美国德克萨斯州
(72)发明人 K·柏林 G·特塞尔 L·波罗
C·斯陶布 C·费尔南德斯
B·西尔瓦
(74)专利代理机构 中国专利代理(香港)有限公司 72001
代理人 张凌苗 张涛
(51)Int.Cl.
G06F 13/10(2006.01)

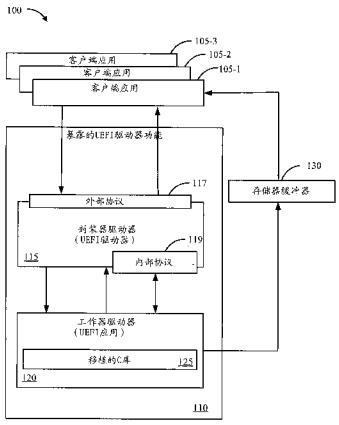
权利要求书2页 说明书5页 附图3页

(54)发明名称

统一可扩展固件接口(UEFI)驱动器和协议

(57)摘要

一种示例装置可以包括处理器和包括计算机程序代码的存储器设备。存储器设备和计算机程序代码利用处理器可以使所述装置：运行客户端应用，所述客户端应用使用第一协议，所述协议已经由统一可扩展固件接口(UEFI)封装器驱动器产生；利用客户端应用，调用UEFI封装器驱动器以执行协议的至少一个操作；利用封装器驱动器加载工作器应用的二进制映像以调用至少一个操作。工作器应用调入软件库的至少一个功能以执行至少一个操作。



CN 105378689 A

1. 一种装置，包括：

处理器；和

包括计算机程序代码的存储器设备，存储器设备和计算机程序代码利用所述处理器，以使所述装置：

运行客户端应用，所述客户端应用使用第一协议，所述协议已经由统一可扩展固件接口（UEFI）封装器驱动器产生；

利用客户端应用调用 UEFI 封装器驱动器以执行所述协议的至少一个操作；

利用封装器驱动器加载工作器应用的二进制映像以调用至少一个操作，

其中，工作器应用调入软件库的至少一个功能以执行至少一个操作。

2. 根据权利要求 1 所述的装置，其中，所述计算机程序代码还使工作器应用经由由 UEFI 封装器驱动器产生的第二协议来查询 UEFI 封装器驱动器，和从 UEFI 封装器驱动器接收参数，所述参数被用于利用至少一个功能来执行至少一个操作。

3. 根据权利要求 2 所述的装置，其中，所述计算机程序代码还使至少一个功能基于所述参数来执行至少一个操作。

4. 根据权利要求 1 所述的装置，其中，所述计算机程序代码还使工作器应用将输出数据写到存储器缓冲器，所述输出数据从至少一个操作得出。

5. 根据权利要求 4 所述的装置，其中，所述计算机程序代码还使客户端应用从存储器缓冲器检索输出数据。

6. 一种方法，包括：

经由由统一可扩展固件接口（UEFI）封装器驱动器产生的第一协议来接收 UEFI 封装器驱动器处的调用，所述调用指示以执行至少一个操作；

加载工作器应用的二进制映像；

经由由 UEFI 封装器驱动器产生的第二协议接收来自工作器应用的二进制映像的查询；

响应于所述查询，从 UEFI 封装器驱动器向工作器应用发送执行至少一个操作所需的参数，和

利用工作器应用调入软件库的至少一个功能，以基于所述参数执行至少一个操作，

其中，UEFI 封装器驱动器、工作器应用和至少一个功能在处理器上运行。

7. 根据权利要求 6 所述的方法，还包括：

运行至少一个功能以执行至少一个操作。

8. 根据权利要求 6 所述的方法，其中，所述至少一个功能是 C 程序。

9. 根据权利要求 6 所述的方法，还包括：

利用工作器应用，在至少一个功能执行至少一个操作之后，将输出数据写到存储器缓冲器。

10. 根据权利要求 9 所述的方法，其中，从使用第一协议的客户端应用接收调用，所述方法还包括：

利用客户端应用从存储器缓冲器检索输出数据。

11. 一种体现在非临时性计算机可读介质上的计算机程序产品，包括：

运行统一可扩展固件接口（UEFI）封装器驱动器的计算机代码，所述 UEFI 封装器驱动

器产生第一协议和第二协议；

运行客户端应用的计算机代码，所述客户端应用使用第一协议以调用第一协议的至少一个操作，

响应于至少一个操作被调用而利用 UEFI 封装器驱动器加载工作器应用的二进制映像的计算机代码；和

运行工作器应用的计算机代码，所述工作器应用调入软件库的至少一个功能以执行至少一个操作。

12. 根据权利要求 11 所述的计算机程序产品，其中，运行工作器应用的计算机代码使工作器应用经由第二协议针对执行至少一个操作所需的参数而查询 UEFI 封装器驱动器应用。

13. 根据权利要求 12 所述的计算机程序产品，其中，运行 UEFI 封装器驱动器的计算机代码使 UEFI 封装器驱动器记录关于存储器中的至少一个操作的详情，以及响应于所述查询而向工作器应用发送详情的至少一部分。

14. 根据权利要求 11 所述的计算机程序产品，其中，当至少一个调入的功能执行至少一个操作时，退出工作器应用，并且控制返回给 UEFI 封装器驱动器，并且运行 UEFI 封装器驱动器的计算机代码还使 UEFI 封装器驱动器加载工作器应用的另一个二进制映像以执行另一个调用的操作。

15. 根据权利要求 14 所述的计算机程序产品，其中，运行工作器应用的计算机代码使工作器应用的第二二进制映像调入软件库的另一个功能，以执行另一个调用的操作。

统一可扩展固件接口（UEFI）驱动器和协议

背景技术

[0001] 基本输入/输出系统(BIOS)是可以在软件基础设施方面提供有限环境的固件接口，其可用于与个人计算机和其他计算设备对接。在采用统一可扩展固件接口(UEFI)标准以取代或补充BIOS的情况下，已经出现固件级别的基于软件的差异化的新的机遇。

附图说明

[0002] 为了更完整地理解各种示例，现在参考结合附图考虑的以下描述，其中：

[0003] 图1图示用于在UEFI环境中利用C库功能的示例运行流程；

[0004] 图2图示示例系统；和

[0005] 图3图示用于在UEFI环境中利用C库功能的示例过程。

具体实施方式

[0006] 统一可扩展固件接口(UEFI)标准可以提供用于通过添加针对新应用和驱动器的开发的灵活性来在BIOS环境上进行改进的框架。然而，与其他开发环境相比，在系统引导时可用的软件基础设施（例如，库和工具）在当前UEFI环境的情况下可能相对较差。

[0007] UEFI是定义操作系统和平台固件及硬件之间的软件接口的规范。UEFI可取代在较旧的个人计算机(PC)模型中使用的基本输入/输出系统(BIOS)固件接口。虽然BIOS被实现为固件片，但UEFI可包括置于计算机固件的顶层的可编程软件接口（包括例如可以将类似功能（例如，预引导或自展(bootstrap)功能）执行为“传统(legacy)”BIOS的基于UEFI的BIOS类固件）和硬件。

[0008] UEFI可以提供称为EFI字节代码或EBC的设备驱动器环境。系统固件可以包括针对驻留在或加载到EBC环境中的任何EBC映像(image)的解释器。EBC仅是UEFI模块可被编译成的一种可能的目标二进制格式。UEFI驱动器例如可被编译成不是EBC的格式。UEFI驱动器可以提供针对硬件组件的支持并执行针对客户端应用的任务。UEFI可以定义能够运行UEFI应用的壳(shell)环境。壳环境通常用作开发/诊断/支持环境。相比而言，UEFI BIOS可潜在的具有许多UEFI应用，以执行各种用户可配置任务，例如以配置比如引导设备偏好等的PC设置。UEFI可以定义作为用于在两个二进制模块（例如，驱动器或客户端应用）之间通信的软件接口集的协议。UEFI驱动器可以经由协议向其他驱动器和应用提供服务。协议可以由UEFI驱动器产生，并且可以被其他UEFI驱动器和任何类型的UEFI应用使用(consume)。

[0009] UEFI中更健壮的特征的实现方式可要求程序员充分开发或适配库、应用和/或驱动器，以符合UEFI的编程模型和应用编程接口(API)，其虽然基于C，但可能不符合现有的C标准（例如，ANSI C、C99等）。为了弥补该差距的部分，根据C95规范的标准C库的端口以及附加地伯克利软件分发(BSD)套接字(socket)库的端口以及一些可移植操作系统接口(POSIX)功能可被包括在开源UEFI开发套件（又名TianoCore）中，作为“UEFI应用开发套件”(EADK)的一部分。

[0010] 由于关于平台（如小型闪存 /ROM 部件）的存储空间约束，固件开发中的另一个挑战可以是保持固件二进制映像的大小较小。在一些 C 开发环境中，使用动态链接（或共享）库 (DLL) 是解决该问题的常用方法。利用该方法，可以在目标环境中安装库二进制的仅一个拷贝，而不管使用库的应用的数目（因此仅占用与库二进制大小一样多的字节）。在 UEFI 环境中，动态链接可通过使用 UEFI 驱动器实现，其中 UEFI 驱动器可以实现将被 UEFI 应用或其他 UEFI 驱动器使用（例如，链接）的库，并且可以通过一个或多个 UEFI 协议接口暴露它。

[0011] 由 UEFI 应用开发套件提供的标准 C 库的使用可仅限于从 EFI 壳启动的 UEFI 应用（例如，“托管的实现方式”）。另一方面，UEFI 驱动器可能无法直接利用 (leverage) 这些 C 库。在本文描述的各种示例中，用户可以被允许规避该限制和将现有基于标准的 C 库（例如包括，依赖于使用套接字的用于网络通信的标准 C API 的许多现有库）移植 (port) 到 UEFI 环境上，所述 UEFI 环境包括部署 C 库的单个拷贝作为 UEFI 驱动器的能力。

[0012] 图 1 图示根据一个示例的用于在 UEFI 环境中利用 C 库功能的示例运行流程 100。示例运行流程 100 可以包括一个或多个客户端应用 105（在图 1 的示例中图示三个客户端应用 105-1、105-2 和 105-3），其可以调用 (invoke) 一个或多个暴露的 UEFI 驱动器功能 110。客户端应用 105 可以经由用于往来于 UEFI 驱动器功能 110 的通信的一个或多个外部协议 117 被链接到并调用 UEFI 驱动器功能 110。UEFI 驱动器功能 110 可以包括两个组件：封装器 (wrapper) 驱动器 115 和工作器 (worker) 应用 120。封装器驱动器 115 可被实现为常规的 UEFI 驱动器（例如，提供加载 / 卸载能力、安装协议等）。工作器应用 120 可被实现为 UEFI 应用。在各种示例中，工作器应用 120 可包含通过从标准的基于 C 的库 125 移植的各种功能提供的核心功能，其可以与工作器应用 120 静态链接。示例运行流程 100 可以采用允许工作器应用 120 被嵌入在封装器驱动器 115 的二进制映像中和从其内加载的技术，从而允许封装器驱动器 115 和工作器应用 120 被捆绑成单个二进制映像。

[0013] 在运行流程 100 的各种示例中，封装器驱动器 115 可以隐藏工作器应用 120，并且可以通过一个或多个外部协议 117 仅仅暴露去往工作器应用 120 的接口，其包括在工作器应用 120 中实现无论什么特征。经由外部协议 117 暴露的特征进而由封装器驱动器通过一个或多个内部协议 119 与工作器应用连接，以命令工作器应用执行实际工作。在各种示例中，封装器驱动器 115 可将工作委托给工作器应用 120。如上所讨论的，协议可基本上是由要被其它系统组件（例如，类似于公布的 DLL 或共享库的 API）所使用的 UEFI 驱动器公布的功能指针和数据结构 (API) 的块。通过在工作器应用 120 中实现核心功能并经由内部协议 119 调用 C 库特征，示例性运行流程 100 可以能够规避技术限制，该技术限制可防止构建与由 UEFI 开发套件 (EADK) 提供的标准 C 库直接链接的 UEFI 驱动器。

[0014] 在各种示例中，工作器应用 120 可以被存储在用于封装器驱动器 115 的代码内。在这方面，每次加载工作器应用 120 时，它可以从存储器中的固定位置访问。因此，可以减少或消除审查 ROM 中的驱动器文件或将工作器应用从 ROM 加载到 RAM 的需要。

[0015] 在各种示例中，工作器应用 120 可以代表封装器驱动器 115 运行操作。为此，在一些示例中，当由外部协议 117 调入 (call) 时，封装器驱动器 115 可以加载和启动工作器应用 120；当被启动时，工作器应用可以使用内部协议（本文称为 GetOperation）查询驱动器以确定什么操作将被工作器应用 120 运行。在执行 GetOperation 查询之后，工作器应用可

以利用由封装器驱动器 115 提供的参数调入在移植的 C 库 125 中的（一个或多个）对应功能。在一些示例中，在执行指定的操作之后，工作器应用 120 可以在存储器缓冲器 130 中存储任何结果，并且然后可以退出，将控制返回给封装器驱动器 115，所述封装器驱动器可以然后在完成由客户端应用 105 之一请求的任务时将控制返回给客户端应用 105。客户端应用然后可以从存储器缓冲器 130 检索结果。以这种方式，对于由客户端应用 105 向通过封装器驱动器 115 暴露的外部 API 进行的每次调入，工作器应用 120 可被加载到存储器内、被运行和被卸载。

[0016] 现在参考图 2，图示示例系统 200。例如，图 2 的示例系统 200 可以实现图 1 的运行流程 100。示例系统 200 可以包括耦合到 UEFI 软件接口 230 的操作系统 210，所述 UEFI 软件接口 230 可以耦合到处理器 250。示例处理器 250 可以是可处理电子指令的任何处理器，诸如芯片或芯片组。在各种示例中，处理器 250 可具有非临时性存储器设备 252。示例存储器设备 252 可以包括各种类型的存储设备中的任一种，诸如闪速存储器、硬盘或其它这样的存储设备。在各种示例中，存储器设备 252 可以与处理器 250 整体地形成或者可以是外部存储器设备。存储器设备 252 可以包括可以由处理器运行的程序代码。例如，可以执行一个或多个过程以运行如以上参考图 1 描述的客户端应用 105。

[0017] 在各种示例中，UEFI 软件接口 230 可以包括预引导模块 235，其提供了 UEFI 预引导环境。预引导环境允许 UEFI 应用（和驱动器）作为系统自举序列的部分运行，其可包括自动加载预定义 UEFI 模块集（驱动器和应用）。作为自动加载的替代方案，在操作系统 210 引导之前，自举序列或其一部分可以由用户干预（例如，通过按压键盘上的键）触发。在各种示例中，待加载的模块的列表可以被硬编码到系统 ROM 中。例如，BIOS 设置菜单用户接口可以是一个这样的应用。

[0018] 示例系统 200 还可以包括客户端应用数据库 220，其可以存储用于图 1 的示例的客户端应用 105-1、105-2 和 105-3 的计算机程序代码。在客户端应用数据库中的客户端应用 105 可以调用由 UEFI 软件接口 230 实现的 UEFI 驱动器。客户端应用 105 可包括在由预引导模块 235 调用的自举序列期间使用的应用。在各种示例中，UEFI 驱动器可以包括封装器驱动器，诸如上述封装器驱动器 115，其可以调用工作器应用，诸如上述工作器应用 120。封装器驱动器和工作器应用可以被存储在客户端应用数据库 220 中。

[0019] UEFI 软件接口 230 可以与存储标准 C 库（诸如由 UEFI 应用开发套件提供的标准 C 库）的 C 库数据库 340 进行通信。如上所述，响应于执行经由内部协议 119 对封装器驱动器 115 之一的 GetOperation 查询，工作器应用 120 可以动态链接到 C 库数据库 240 中的各种功能。

[0020] 图 3 图示用于在 UEFI 环境中利用 C 库功能的示例过程 300，例如，如上面参考图 1 所描述。在各种示例中，过程 300 可至少部分由图 2 的示例系统 200 执行。过程 300 将进一步参考图 1 和 2 来描述。

[0021] 在图 3 中图示的示例中，过程 300 可以以通过处理器 250 例如结合 UEFI 软件接口 230 和预引导模块 235 发起客户端应用 105 之一而开始。客户端应用 105 可以被存储在客户端应用数据库 220 中。客户端应用 105 可被编写成使得其使用由封装器驱动器 115 产生的 UEFI 协议（例如，在图 1 中的外部协议 117），并且可以调用 UEFI 协议的操作（框 310）。处理器 250 可以发起运行封装器驱动器 115，以便调用所使用的协议的操作。

[0022] 当被发起时，封装器驱动器 115 可以使处理器 250 记录关于所调用的操作的详情（框 315）。例如，可以在存储器设备 252 中内部记录所述详情。在各种示例中，所记录的详情可包括要运行哪些功能，要使用哪些输入参数以及要在运行调用的操作之后返回哪些输出数据。

[0023] 在框 320 处，封装器驱动器 115 可以使处理器 250 将工作器应用 120 的二进制映像加载到存储器 252 中。在各种示例中，可以使用 UEFI 接口 230 的 LoadImage（加载映像）系统服务功能来加载工作器应用 120 的映像。处理器 250 然后可以开始运行工作器应用 120 的二进制映像。这可以使用 UEFI 接口 230 的 StartImage（开始映像）系统服务功能实现。开始运行工作器应用 120 可阻止客户端应用 105 调入封装器驱动器 115，直到工作器应用 120 运行、退出并返回到封装器驱动器 115 为止。

[0024] 当在框 320 处由封装器驱动器 115 发起时，工作器应用 120 可使用内部协议 119 查询封装器驱动器 115，所述内部协议 119 提供上述 GetOperationAPI（框 325）。在图 3 的示例中，在框 330 处，响应于该查询，封装器驱动器 115 可返回对于运行操作必要的参数，其在阶段 315 由封装器驱动器 115 记录。在各种示例中，工作器应用 120 可接收参数（框 330）。

[0025] 在各种示例中，工作器应用 120 可基于在框 330 处从封装器驱动器 115 接收的参数运行来自 C 库数据库 240 的一个或多个移植的功能（框 335）。当接收到功能时，工作器应用 120 可以使处理器 250 使用在框 330 处获得的输入参数来运行移植的功能。

[0026] 在图 3 的示例中，在框 340 处，在完成移植的 C 功能的运行之后，工作器应用 120 可以将输出数据写到存储器缓冲器 130。在一些示例中，在 C 库中，例如，数据可通过输出指针返回到调入者。这由 UEFI 接口 230 完全支持，其利用如下事实：在引导服务可用的同时在 UEFI 中没有过程 / 存储器隔离。尽管客户端应用 105、封装器驱动器 115 和工作器应用 120 都是独立的可执行二进制映像这一事实，来自 C 库数据库的底层功能可以能够写到由上层（例如，客户端应用 105 和 / 或封装器驱动器 115）传下来的存储器地址。

[0027] 在图 1 的示例过程的框 345 处，工作器应用 120 可退出运行，并且控制可经由 StartImage 系统服务功能返回给封装器驱动器 115。在框 350 处，如果其他操作需要被调用来完成在框 315 处记录的详情，可以重复框 320-345 的处理，直到所有操作被完成为止。当完成所有操作时，封装器驱动器 115 可以退出，并且控制可被返回给客户端应用 105（框 355）。在框 360 处，例如，客户端应用 105 可以在存储器缓冲器 130 中在指定的缓冲器地址处检索由工作器应用 120 存储的输出数据。

[0028] 在图 3 中图示的过程 300 仅是示例而不是限制性的。在各种示例中，过程 300 可以例如通过使步骤或块被添加、移除、重布置、组合和 / 或并发执行而被变更。例如，在一些示例中，框 360（其中，客户端应用 105 可从存储器缓冲器 130 检索输出数据）可在框 320-345 被重复之前执行（见框 350），以在阶段 310 处执行由客户端应用调用的其它操作。对如示出和描述的过程 300 的又其它的变更是可能的并被预计在本公开文件的范围内。

[0029] 上文描述的方法和系统的一个潜在的益处是减少的开发时间和因此上市时间以及基于 UEFI 的固件特征的提高的质量，允许在 UEFI 环境中重用现有的经证实的基于 C 的应用 / 库。

[0030] 本文描述的各种示例在方法步骤或过程的一般上下文中描述，其可以在一个示例中通过体现在机器可读介质中的软件程序产品或组件来实现，所述软件程序产品或组件包

括可执行指令，诸如由联网环境中的实体运行的程序代码。通常，程序模块可以包括例程、程序、对象、组件、数据结构等，其可被设计成执行特定任务或实现特定抽象数据类型。可执行指令、关联的数据结构和程序模块表示用于执行本文公开的方法的步骤的程序代码的示例。这样的可执行指令或关联的数据结构的特定序列表示用于实现在这样的步骤或过程中描述的功能的对应动作的示例。

[0031] 各种示例的软件实现方式可以利用具有基于规则的逻辑和其他逻辑的标准编程技术来实现，以实现各种数据库搜索步骤或过程、相关步骤或过程、比较步骤或过程和决策步骤或过程。

[0032] 已经出于说明和描述的目的呈现各种示例的前述描述。前述描述并不旨在是穷尽的或限制于所公开的示例，并且修改和变化根据上述教导而是可能的，或者可以从各种示例的实践中获得。本文所讨论的示例被选择和描述，以便解释本公开的各种示例的原则和性质和其实际应用，以使得本领域技术人员能够利用各种示例中的和具有如适合于预计的特定使用的各种修改的本发明。本文所描述的示例的特征可以以方法、装置、模块、系统和计算机程序产品的所有可能的组合进行组合。

[0033] 本文还注意到，虽然以上描述了示例，但是这些描述不应当被视为限制意义。而是，存在在不背离如所附的权利要求中限定的范围的情况下可以做出的若干变化和修改。

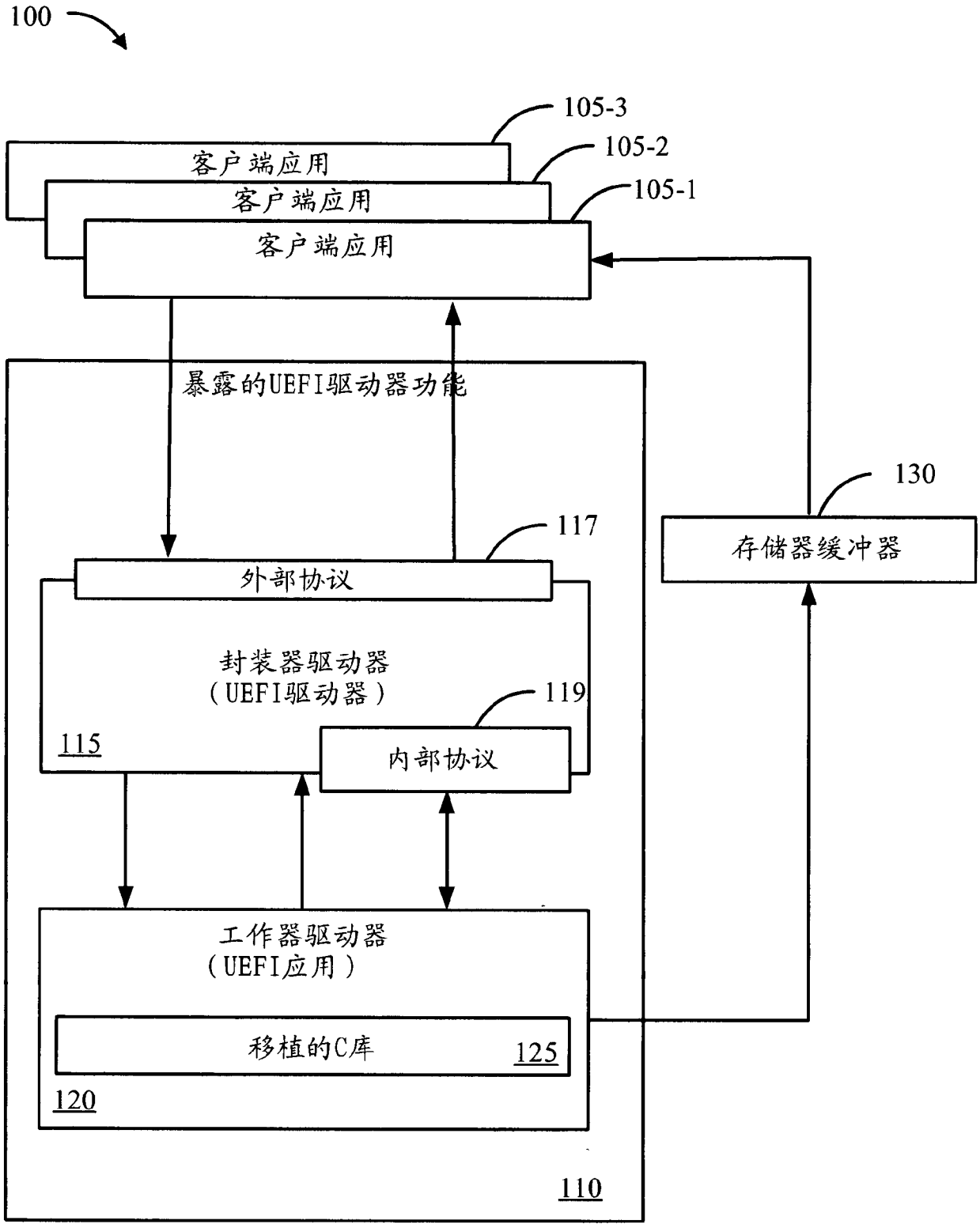

100
105-3
客户端应用
105-2
客户端应用
105-1
客户端应用
暴露的UEFI驱动器功能
130
存储器缓冲器
117
外部协议
封装器驱动器
(UEFI驱动器)
119
115
内部协议
工作器驱动器
(UEFI应用)
移植的C库
125
120
110

图 1

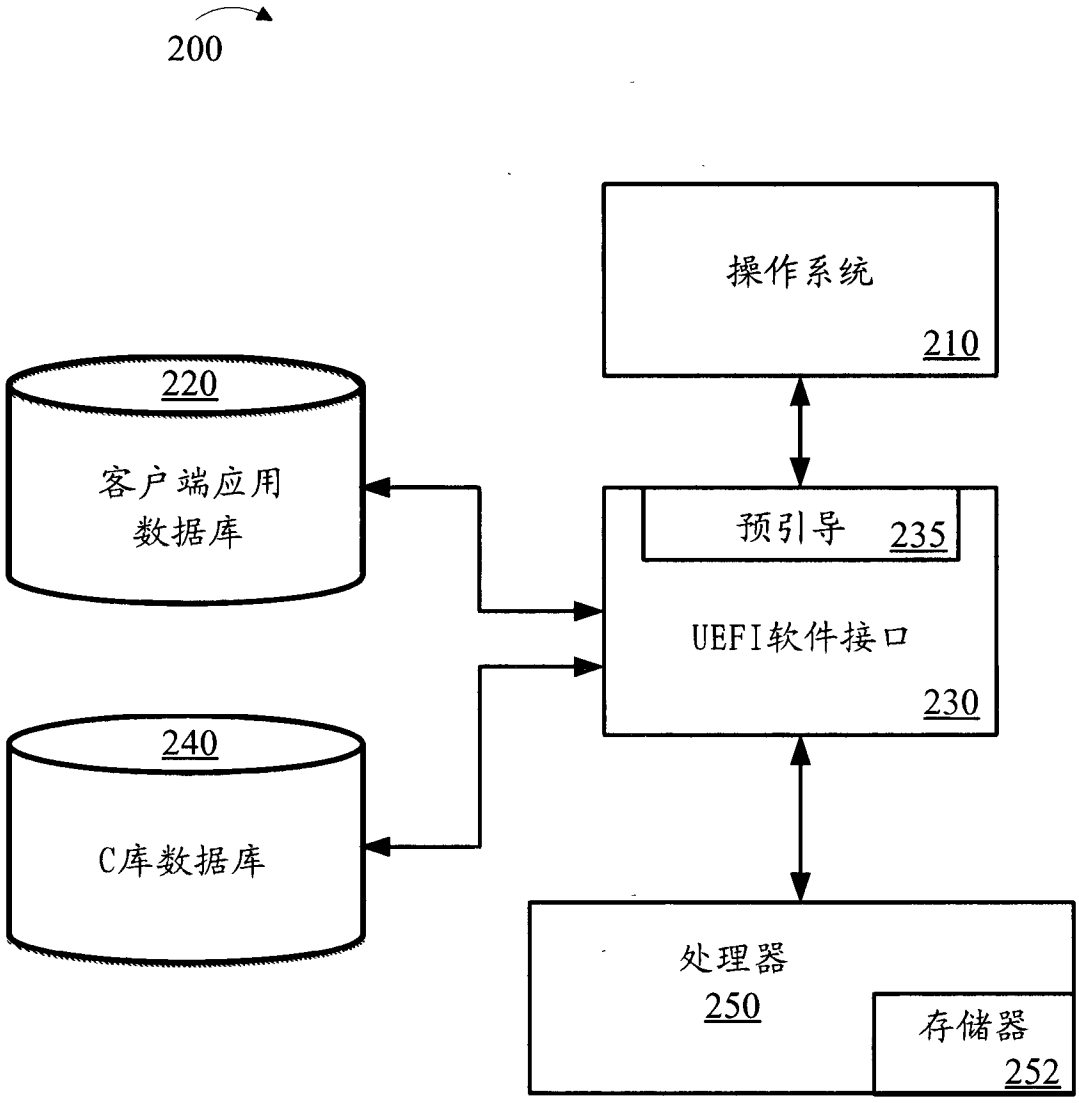

200
操作系统
210
220
客户端应用
数据库
预引导
235
UEFI软件接口
230
240
C库数据库
处理器
250
存储器
252

图 2

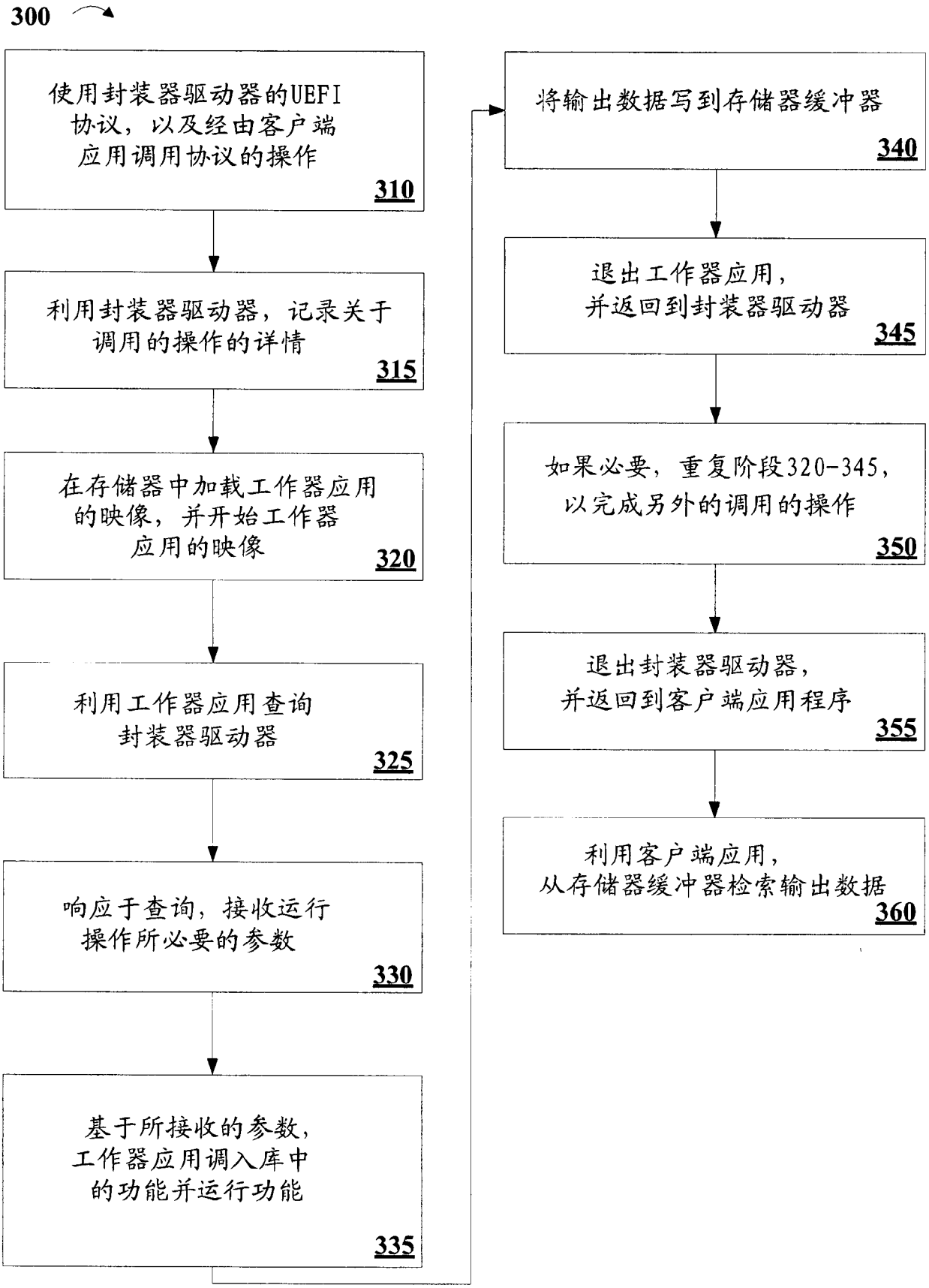

300
使用封装器驱动器的UEFI协议，以及经由客户端应用调用协议的操作
310
利用封装器驱动器，记录关于调用的操作的详情
315
在存储器中加载工作器应用的映像，并开始工作器应用的映像
320
利用工作器应用查询封装器驱动器
325
响应于查询，接收运行操作所必要的参数
330
基于所接收的参数，工作器应用调入库中的功能并运行功能
335
将输出数据写到存储器缓冲器
340
退出工作器应用，并返回到封装器驱动器
345
如果必要，重复阶段320-345，以完成另外的调用的操作
350
退出封装器驱动器，并返回到客户端应用程序
355
利用客户端应用，从存储器缓冲器检索输出数据
360

图 3