

[19] 中华人民共和国国家知识产权局



[12] 发明专利说明书

专利号 ZL 200580033195.9

[51] Int. Cl.

G06F 11/30 (2006.01)

G06F 11/14 (2006.01)

[45] 授权公告日 2009 年 2 月 11 日

[11] 授权公告号 CN 100461128C

[22] 申请日 2005.9.29

CN1347037A 2002.5.1

[21] 申请号 200580033195.9

审查员 刘清泉

[30] 优先权

[74] 专利代理机构 中国专利代理(香港)有限公司

[32] 2004.9.30 [33] US [31] 10/957,444

代理人 刘 红 梁 永

[86] 国际申请 PCT/US2005/035373 2005.9.29

[87] 国际公布 WO2006/039593 英 2006.4.13

[85] 进入国家阶段日期 2007.3.30

[73] 专利权人 英特尔公司

地址 美国加利福尼亚州

[72] 发明人 M·卡塔里亚 A·加夫肯

W·小斯蒂芬斯

[56] 参考文献

US6732267B1 2004.5.4

权利要求书 3 页 说明书 19 页 附图 4 页

US6708231B1 2004.3.16

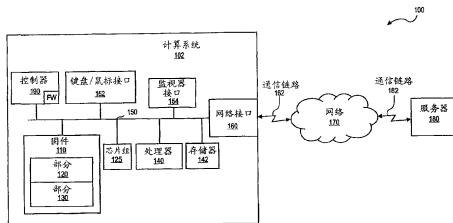
US5938764A 1999.8.17

[54] 发明名称

通过网络对固件的自监控和更新

[57] 摘要

实施例包括监控计算系统以确定该计算系统的固件是否被破坏、挂起或者需要自动更新。该计算系统可以随后通过网络请求固件更新数据。此外，该计算系统可以包括具有确定固件是否被破坏或挂起并通过网络请求和接收固件更新数据的能力的控制器。另外，该控制器可以具有在固件被破坏或挂起、处理器被挂起、操作系统被停止、挂起或软关机时运行的功能。另外，如果该控制器检测到固件被破坏或挂起，该控制器可以在更新固件时停止处理器。



1、一种装置，包括：

处理器，用来运行操作系统；

固件，具有包括配置数据的第一部分和第二部分，第一部分包括用来监控第二部分以确定第二部分是否被破坏的逻辑；和

控制器，包括固件以便通过网络向服务器发送消息以请求更新的配置数据，其中该消息是为了从服务器的存储器中存储的数据库请求更新的配置数据、通过网络接收更新的配置数据、并用返回的更新的配置数据重写第二部分的至少一部分，其中该消息包括包含该处理器和固件的计算系统的标识；

其中第一部分包括监控时钟信号的逻辑，以确定是否所选择的时间或者是否一段时间已经过去表明需要更新配置数据，并且其中第一部分激活用来发送预定的通知信号的硬件部分、用来发送预定的通信软件通知消息给控制器的软件部分之一，以便向控制器表明计算系统的引导是否已经被启动。

2、权利要求 1 的装置，其中第一部分包括固件，以监控客户机芯片组以比较处理器的进度和根据配置数据的处理器的期望响应。

3、权利要求 1 的装置，其中第一部分与第二部分相耦合，并且包括监控第二部分以确定第二部分是否被停止或挂起的逻辑。

4、权利要求 3 的装置，其中第一部分包括当第二部分被破坏、停止或挂起时停止处理器的逻辑。

5、权利要求 4 的装置，其中第一部分包括在如果第二部分被破坏、停止或挂起时用于执行的电路或固件。

6、权利要求 1 的装置，其中第一部分被存储在受擦除保护的存储器中，该存储器包括硬编码的系统参数以发送表示第二部分遭破坏的预定消息给控制器。

7、权利要求 6 的装置，其中控制器包括嵌入式微控制器、固件和电路之一。

8、权利要求 6 的装置，其中第一部分包括对第二部分执行基于校验和的检查、基于签名的检查和基于硬件的检查之一的逻辑。

9、权利要求 1 的装置，其中所述逻辑包括电路、固件、软件代码、

数字数据和软件之一。

10、一种计算机实现的方法，包括：

监控计算系统以确定该计算系统的固件是否被破坏；

如果该部分被破坏，从该计算系统通过网络请求更新数据以更新固件中被破坏的部分；

监控系统的进度，

检测根据固件的预期响应和进度之间的不一致性；

检测在计算系统的设备执行的功能和用固件中存储的设备配置确定的该设备的预期响应之间的差异；

如果确定固件被破坏，使计算系统的处理器暂停处理；

如果(1)固件被挂起或停止或(2)处理器被暂停，则更新遭破坏的固件；和

其中更新包括在所选择的时间上或者在所选择的时间间隔之后请求固件的固件更新数据，其中请求更新包括在(1)固件被挂起或停止后或(2)处理器被暂停后请求更新。

11、权利要求10的计算机实现的方法，其中监控包括：

执行基于校验和的检查、基于签名的检查和基于硬件的检查之一。

12、权利要求10的计算机实现的方法，其中监控包括：

如果固件被挂起或停止，确定固件被破坏。

13、一种计算机实现的方法，包括：

通过网络连接接收固件更新数据；

用该固件更新数据更新计算系统的固件；

其中如果计算系统的固件被停止则进行接收和更新；

其中更新包括在所选择的时间上或者在所选择的时间间隔之后请求固件的固件更新数据；

其中该固件包括第一部分和第二部分，并且更新包括用由固件的第一部分用通过网络请求到的更新的第二部分数据更新固件的第二部分；

其中更新包括更新固件以消除固件的功能或进程中的错误、向固件添加功能或进程、以及增强固件的功能或进程之一，和

其中所述所选择的时间或所选择的时间间隔是由管理员选择的，或者是在计算系统的部分的制造期间选择的。

14、权利要求13的计算机实现的方法，其中更新包括：

用通过网络请求到的更新的第一部分数据更新固件的第一部分；以及随后

用由第一部分请求到的更新的第二部分数据更新固件的第二部分。

15、权利要求 13 的计算机实现的方法，其中更新包括：

用计算系统的唯一标识请求固件更新数据；

通过网络从服务器接收固件更新数据；

将固件更新数据返回给固件，或者将固件更新数据重写到固件。

16、权利要求 13 的计算机实现的方法，其中接收包括使用安全服务器会话和使用加密数据之一。

通过网络对固件的自监控和更新

技术领域

更新和恢复系统固件。

背景技术

在电子设备(例如计算系统)的初始化和/或使用期间, 固件“程序”通常被用来检测、测试、初始化以及监控设备或系统硬件。例如, 当一台个人计算机(PC)被打开时, 通常基本输入/输出系统(BIOS, basic input/output system)程序(例如固件)会被运行, BIOS是存储在或编程到PC主板中的只读存储器(ROM)芯片(一般称为BIOS芯片)的软件。一旦被运行, BIOS立即识别所有的板上设备(例如处理器、存储器、视频卡等)然后运行加电自检(POST, power-on self-test)以确定各个设备是否工作正常。如果所有设备都通过了POST测试, 然后BIOS初始化各个设备、检测硬盘驱动器、光盘(CD-ROM)驱动器以及软盘驱动器。然后BIOS在第一引导设备(通常是硬盘驱动器或软盘驱动器)上查找加载操作系统软件所必需的文件。BIOS可以处理到键盘和显示控制器等外设控制器的低层输入/输出(I/O)。此外, BIOS将检测、测试、初始化和监控系统硬件直到操作系统接管它们。然后, BIOS将对系统的控制传给操作系统软件。如果有任何设备没有通过POST, 将会有错误消息显示在屏幕上, 或者有一系列“滴滴声”(beep)通过PC扬声器播放出来以表示有问题存在。滴滴声的次序(滴滴声代码)可以用来标识存在的问题的类型。

但是, BIOS或固件通常是被“烧制”到ROM芯片的一个部分中, 并被写进或编程到ROM芯片的第二部分中。尽管“烧制”的部分可以被硬件机制(例如, 紫外线辐射)“闪光(flash)”擦除, 但第二部分可以由软件机制擦除或写入。因此, BIOS或固件可能被破坏、变得不准确或者被“病毒”、“蠕虫”或“黑客”修改。此外, 计算系统的用户可能无意中将BIOS修改成不合需要的配置。因而, 能够恢复和更新BIOS或固件以保证它正确非常重要。

附图说明

图 1 是通过网络自监控和更新固件的系统的结构图；

图 2 是通过网络更新固件的过程的流程图；

图 3 是通过网络更新固件的过程的流程图；

图 4 是通过网络自动更新固件的过程的流程图。

具体实施方式

图 1 是通过网络自监控和更新固件的系统的结构图。图 1 所示计算环境 100 包括通过通信链路 162 与网络 170 相连的计算系统 102 以及通过通信链路 182 与网络 170 相连的服务器 180。计算设备 102 可以是客户计算机、服务器计算机、台式计算机、膝上型计算机、个人数字助理 (PDA)、蜂窝电话或任何其它有操作系统和固件的数字处理器或处理系统。

如图 1 中所示，计算系统 102 包括通过网络接口 160 与通信链路 162 相连的总线 150。总线 150 可以是计算机总线，如主板上用于接合计算机的不同部件的总线。网络接口 160 可以是用于连接到 LAN (局域网)、Intranet (企业内部网)、Internet (互联网) 的网络接口，或者另一电子设备或计算系统通信支持网络。相应地，通信链路 162 和网络 170 可以支持在这些网络上的通信。具体地说，例如，网络 170 可以是 LAN、Intranet 或 Internet。还可以预期，网络 170 可以包括多于一种类型的网络和/或技术以提供电子设备和/或计算系统之间的通信。具体地说，网络 170 可以包括无线通信、蜂窝通信、LAN 通信以及 Internet。通信链路 182 可以是类似于上面为通信链路 162 所描述的链路。服务器 180 可以是网络服务器、与计算机系统 102 类似的计算系统或者别的计算机服务器以提供这里所说明的功能。

计算系统 102 示出为还包括与总线 150 相连的键盘/鼠标接口 152。键盘/鼠标接口 152 可以是用于将键盘和/或鼠标连接到计算机系统 102 以向那里提供输入的接口。还示出了监视器接口 154 与总线 150 相连。监视器接口 154 可以提供接口或适当的信号给与计算系统 102 相连的屏幕或监视器。例如，监视器接口 154 可以是视频或显示卡。

图 1 还示出了与总线 150 相连的存储器 142。存储器 142 可以代表随机访问存储器(RAM)、记忆体存储器、通用串行总线(USB, Universal

Serial Bus)存储器、硬盘驱动器、CD-ROM、光盘和/或软盘。存储器 142 可以是机器可访问介质，例如用于包含指令的介质，所包含的指令在被处理器执行时使计算系统 102 执行与计算机有关的任务，包括这里所说明的那些任务，例如通过网络 170 与其它计算系统和/或服务器 180 通信。

还示出了与总线 150 相连的处理器 140。处理器 140 可以是中央处理单元(CPU)、数字信号处理器或其它处理器。能够理解，处理器 140 可以是用于处理由存储器 142 提供的或存储在其中的指令的处理器，例如使计算系统 102 执行与计算机有关的任务，包括这里所说明的那些任务，例如通过网络 170 与其它计算系统和/或服务器 180 通信。处理器 140 可以执行或运行存储器 142 中存储的或可以从它获取的操作系统。因而，计算系统 102 可以是能够通过网络 170 访问服务器 180 的客户计算机或客户机系统 (client system)。

图 1 还示出了与总线 150 相连的固件 110。固件 110 可以是计算机基本输入/输出系统 (BIOS)，客户机系统固件(CSF)、或用于测试、初始化以及加载计算系统 102 的引导程序或操作系统的其它指令。固件 110 可以被存储在存储器中，例如只读存储器(ROM)、非易失性存储器、可擦可编程只读存储器(EPROM)、电子可擦可编程只读存储器(EEPROM)或闪存。例如，固件 110 可以被存储在当电源被从计算系统 102 移除或关闭 (硬关闭) 时不会被擦除的存储器中。另外，固件 110 可以被存储在通过暴露在紫外线中或用紫外线照射能够擦除的存储器中，以更新、恢复或用后续数据重写，例如，通过“烧制”进存储器中。还可以预期用其它过程、固件、软件和/或硬件更新固件 110。

如果固件被破坏就必须要进行固件 (例如 BIOS 和固件 110) 的更新和恢复。如果固件具有与计算系统的设备相比不准确的配置数据、或者导致操作系统或处理器停止 (halt)、挂起或进入软关机 (soft - off) 状态，就可被看作是已被破坏。例如，停止可以发生在硬件停止或其它故障导致操作系统和/或处理器中止固件、软件或操作系统指令的处理时。同样，挂起可以发生在软件循环问题或其它故障导致操作系统和/或处理器中止固件、软件或操作系统指令的处理时。最后，软关机指计算系统被加电但操作系统和/或处理器因为计算系统处在节电模式或别的硬件或软件模式而中止固件、软件或操作系统指令的处理时。

固件还可以被更新或恢复以消除功能或过程中的“bug”或错误、添加功能或过程、或者增强固件的功能或过程(例如，添加新设备的配置)。此外，可能希望在固件被“病毒”、“蠕虫”或“黑客”破坏或更改后更新或恢复固件。有些情况下还可能希望在例如所选择的时间上或周期性地自动更新固件以确保固件持续准确或对这里所说明的固件提供更新。

图 1 还示出了与总线 150 相连的控制器 190。控制器可以包括计算机芯片、数字信号处理器、嵌入式微控制器、固件、电路、计算机硬件和/或其中存储的计算机软件。特别地，所示控制器 190 具有固件 FW。控制器 190 可以是有诸如代码、程序设计、固件 FW、数据、智能和/或计算机软件等“逻辑”以执行这里所说明的任务的智能网络控制器 (INC, Intelligent Network Controller)。通常，这里所说明的代码、程序设计、数据、智能、计算机软件、固件、硬件和/或计算机硬件在某种意义上可以被定义为“逻辑”，所述“逻辑”具有执行关于该“逻辑”所说明的功能的能力。

例如，控制器 190 可以包括监控固件 110 以及与服务器 180 通信的逻辑。更特别地，控制器 190 可以包括用于跨越网络 170 与固件 110 和服务器 180 通信以从服务器 180 获取固件映像 (image) 从而更新固件 110 的逻辑。控制器 190 的逻辑具有在网络上请求和接收固件 110 的最新 (updated) 配置或固件数据的能力。例如，控制器 190 还可以具有用安全服务器会话或加密数据在网络 170 上与服务器 180 进行安全通信的能力。特别地，计算系统 102 和服务器 180 之间在通信链路 162 和 182 以及网络 170 上的通信可以是安全服务器会话，例如事务层安全(TSL, Transactional Layer Secure)服务器会话，和/或使用加密数据(例如 RSA(Ribest-Shamir-Adleman)加密数据)，或者对数据的其它公共/私有密钥加密。控制器 190 还可具有向固件 110 返回接收到的最新配置或固件数据、或者用接收到的最新配置或固件数据重写固件 110 的逻辑。

控制器 190 可以有足够的逻辑以拥有当部分 120 正在执行或被破坏时的功能表现。根据实施例，控制器 190 包括独立于固件 110 执行的逻辑或固件，所以控制器 190 即使在部分 120 被破坏或挂起、处理器 140 被暂停 (held) 或停止、计算系统 102 的操作系统被挂起、停止或软关机时、和/或计算系统 102 被挂起、停止或软关机时也能充分起作用。例

如，当硬件或软件故障导致处理器中止固件、软件或操作系统指令的处理时，处理器 140 可能被暂停或停止。因而，控制器 190 在处理器 140 或计算系统 102 的操作系统被挂起、停止或软关机时能够充分起作用。

此外，控制器 190 可以有足够的逻辑与固件 110 通信以接收固件的有些部分被破坏的指示以及从服务器 180 获取原始的或最新的固件的请求。控制器 190 还可以具有或存储固件或存储器、修正信息、发布日期等信息。

控制器 190、存储器 142 和/或固件 110 可以具有获取固件 110、计算系统 102 的操作系统和/或计算系统 102 自身的标识信息、修正信息、发布日期等信息的逻辑、信息和/或机制。例如，计算系统 102 的标识可以是计算系统 102 在特定公司、特定机构或所处位置或大厦中的类似计算系统之间(例如，其它客户机系统之间)的唯一标识。特别地，控制器 190 可以包括与计算系统 102 的其它部件通信(例如，与处理器 140 和/或固件 110 通信)以根据计算系统 102 的标识(例如唯一标识)识别该计算系统的逻辑。

此外，控制器 190 可以包括逻辑来通过网络 170 将对固件的请求以及计算系统 102 的标识传递给服务器 180、从服务器 180 接收与那些请求有关的响应、并且将接收到的响应发送到固件 110 和/或计算系统 102 的其它部件。例如，控制器 190 可以从部分 120 接收到一个“核心遭破坏”消息、请求更新数据消息或者请求更新固件消息。例如，“核心”可以被定义为固件 110 或其中的一部分。接着，控制器 190 可以将“核心遭破坏”消息、请求更新数据消息或请求固件更新数据消息发送到服务器 180。控制器 190 随后可以等待并从服务器 180 接收固件更新数据并将固件更新数据提供给固件 110。此外，控制器 190 可以包括逻辑，使控制器 190 可以重写或为服务器 180 提供直通 (pass - through) 接口以用从服务器 180 接收到的固件更新数据重写部分 120 和/或部分 130.

另外，计算系统 102 包括芯片组 125。芯片组 125 可以包括具有执行这里关于芯片组 125 所述功能的能力的计算机芯片、数字信号处理器、固件、电路、计算机硬件和/或其中存储的计算机软件、寄存器、缓冲器、计算机逻辑、门电路，等等。在有些情况下，控制器 190 可以通过监控芯片组 125 而监控处理器 140 的进度以监控固件 110 和处理器 140 之间的通信。因而，芯片组 125 允许控制器 190 监控处理器 140 对固件 110

的访问，并从处理器 140 返回数据给固件 110 以检测在计算系统 102 的引导期间是否有错误。特别地，芯片组 125 可以允许控制器 190 监控固件 110 和处理器 140 之间的通信，使得对固件 110 的缺省地址的访问启动系统 102 的引导，以及返回数据通过芯片组 125。

图 1 还示出了通过通信链路 162、182 和网络 170 与计算系统 102 相连的服务器 180。服务器 180 可以是计算机“控制台”，例如被信息技术(IT)管理员监控并控制以控制在某个企业、公司、位置或机构的计算机固件配置的服务器控制台。例如，IT 管理员可以是在服务器 180 上工作或能够访问服务器 180 的人，或者是运行在服务器 180 上或能够访问服务器 180 的自动化应用程序，以提供固件更新给企业、公司、位置或机构的计算机。服务器 180 可以具有包括计算系统(例如计算系统 102)的配置数据和固件更新的记忆体储存器数据库。另外，服务器 180 可以包括操作系统映像。可以通过检索服务器 180 中的数据库访问服务器 180 的固件更新和操作系统映像。或者，可以由管理员(例如，能够访问服务器 180 的人或者运行在服务器 180 上或能够访问服务器 180 的自动化应用程序)从服务器 180 提供固件更新和操作系统映像，例如信息技术(IT)管理员手工从服务器 180 中的数据库选择固件和/或操作系统映像，或者将固件和/或操作系统映像加载到服务器 180 以通过网络 170 进行传播。

在一个实施例中，服务器 180 可以包括标识信息以及相关支持数据(例如固件数据、固件更新、固件映像、操作系统映像等)的数据库。适当的标识信息包括客户机系统标识(例如计算系统 102)(例如，在一个机构的其它客户机系统之间的唯一标识)、操作系统版本、处理器版本、固件修正信息、固件发布日期等。特别地，服务器 180 可以查找、定位并传输对应于从控制器 190 接收到的客户机系统标识的支持数据(例如，对于不同的客户机系统标识，固件修正和/或固件发布日期)或提供对支持数据的访问。在有些情况下，IT 管理员可以访问服务器 180，以使 IT 管理员的动作可以查找、定位和传输支持数据或提供对支持数据的访问。

图 1 示出了包括部分 120 和部分 130 的固件 110。部分 120 和部分 130 可以是固件 110 的整个固件的全部或一部分。根据实施例，部分 120 和部分 130 可以被总称为“核心系统”固件，部分 120 是核心系统固件的部分 1 或“引导块”，部分 130 是核心系统固件的部分 2 或“核心”。

例如，部分 120 可以是由基于硬件的保护和/或基于软件的保护机制进行擦除保护的固件的第一部分。同样，部分 120 可以是被进行了擦除保护以避免被有意或无意擦除的固件 110 的受擦除保护部分或一小部分，只有使用特定的硬件机制才能擦除部分 120，例如用紫外线照射。根据实施例，部分 120 可以代表“烧制”进 ROM、EPROM 或 EEPROM 芯片的一部分中的数据。因而，部分 120，即“烧制”的那部分，可以被硬件机制(例如紫外线照射)“闪光”擦除。

另外，部分 120 可以包括发现和识别系统平台上(如计算系统 102 上)控制器 190 存在的逻辑。例如，部分 120 能够识别系统平台上控制器 190 的硬编码定位 (hard-coded location) 和/或运行时发现 (run-time discovery)。此外，部分 120 可以包括足够的逻辑以发现固件 110 中被破坏的固件，例如使用基于校验和的检查、基于签名的检查和/或基于硬件的检查来确定部分 130 是否被破坏。此外，部分 120 可以包括与控制器 190 通信的逻辑。因此，部分 120 可以通知控制器 190 部分 120 和/或部分 130 被破坏，或者请求更新。例如，部分 120 可以从控制器 190 请求部分 120 和/或部分 130 的原始或最新固件。部分 120 还可以包括足够的逻辑以接收响应这些请求的响应、原始固件、更新固件和/或更新数据，并采取适当的动作，例如用固件或数据更新部分 120 和/或部分 130。

固件 110 还包括部分 130，例如可擦除的固件部分。根据实施例，部分 130 可以代表写入或存储到存储器、ROM、EPROM 或 EEPROM 芯片的一部分中的数据。例如，部分 130 可以被写入或编程到部分 120 被编程进的 ROM 芯片的第二部分中，但部分 130 没有部分 120 的硬件或软件擦除保护。因而，部分 130 可以被软件机制擦除或重写。部分 130 在数据量上可以比部分 120 大得多。部分 130 可以设计用于部分 120 引起的或来自部分 120 的擦除、更新或恢复。部分 120 和部分 130 可以包括计算系统 102 的配置数据。例如，该配置数据可以包括计算系统 102 的设备或与其相连的设备(例如键盘/鼠标接口 152 和监视器接口 154)的配置。预期来自固件 110 的数据可以用来加载计算系统 102 的配置寄存器，例如定义上述设备的运行参数。

因此，根据实施例，计算环境 100 允许固件 110 在挂起、被破坏时或在特定时间被更新，不管计算系统 102 的操作系统是否正常工作。例如，固件 110 可以是驻留在计算系统 102 上的客户机系统固件(CSF)，

客户机系统上的 CSF 具有内置逻辑以发现 CSF 是否被破坏, 还具有“自治愈”或更新遭破坏的 CSF 的能力。

某种情况下, 控制器 190 可以是智能连网控制器(INC), 例如具有支持对遭破坏的固件进行检测并将计算系统 102 带到能够(如通过网络 170)与服务器 180 交互状态的逻辑的控制器。这个逻辑允许控制器 190 获取原始的或最新的固件(例如, 包含提供计算系统 102 的固件功能和/或设备配置信息的软件或编程的代码、数据或映像)以初始写、更新、修复、自治愈、恢复和自更新固件 110 的所有或部分(例如部分 120 和/或部分 130)。

同样, 部分 120 可以是包含足够的逻辑功能以将计算系统 102 带到部分 120 能够与控制器 190 和服务器 180(如, 通过网络 170)交互的状态的 CSF 的一角或部分。这里, 该逻辑允许部分 120 获取原始的或最新的固件以初始写、更新、修复、自治愈、恢复和自更新固件 110 的部分(如部分 130)。可以配置或内置部分 120 的擦除保护以暂停允许部分 130 的恢复(例如, 使用控制器 190 和服务器 180)的逻辑。

相应地, 部分 130 可以是包含计算系统 102 的部件或设备的配置数据的 CSF 的非安全角色或部分。例如, 部分 130 可以是不像部分 120 那么安全的固件的主要部分, 因为部分 130 没有内置与部分 120 类似的硬件或软件机制以保护部分 130 不被擦除。因而, 部分 130 更灵活地允许对部分 130 的故意擦除或字段更新。但应该理解这个相同的灵活性使得部分 130 易受意外擦除。因而, 如果部分 130 被意外擦除, 部分 120 可以提供部分 130 的恢复。

服务器 180 可以是客户机系统(如计算系统 102)可以用之通信以获取适当的固件和/或操作系统映像的控制台。可以通过检索服务器 180 上存储的数据库自动从服务器 180 选择固件和/或操作系统映像。或者, IT(信息技术)管理员可以通过从服务器 180 上的数据库选择正确的固件和/或操作系统映像并控制所选择的固件和/或操作系统映像通过网络 170 与计算系统 102 的通信, 而手动选择固件和/或操作系统映像。

图 2 是通过网络更新固件的过程 200 的流程图。在块 202 系统被引导或重置。例如, 块 202 可以对应于打开、重置、初始化一个计算系统(如计算系统 102)或启动其电源。接着, 在块 203, 控制器被启动或初始化, 与系统继续引导、初始化、运行固件、运行操作系统或处理的能力

无关。例如，块 203 可以对应于控制器 190，控制器 190 正在被启动以初始化和配置自己从而运行固件，随后不管固件 110、部分 120 和/或部分 130 遭破坏或被挂起仍然运行固件。同样，控制器 190 可以执行上述功能而不管处理器 140 是否被停止或暂停；不管计算系统 102 的操作系统是否被停止、挂起或软关机；也不管计算系统 102 是否被停止、挂起或软关机。

在块 205，控制器(如控制器 190)被通知客户机系统启动，例如计算系统 102 的启动。块 205 可以包括发送预定信号给控制器以表明计算系统 102 的引导是否已经被启动(如，在计算系统 102 的电源被打开后或计算系统 102 被“重置”后)的通过硬件机制(例如，通过硬件或电路)的通知。另外，可以用软件机制通知控制器 190 计算系统的启动，例如向控制器 190 发送预定的通信消息以表示计算系统 102 的引导。考虑该通知可以包括通过硬件接口发送预定的通知信号或通过固件 110 和控制器 190 之间的通信接口发送预定的软件消息。

在块 210，监控系统以确定固件是否被破坏。例如，控制器 190 可以包括内置在控制器 190 的固件中的方法或逻辑以监控、确定或发现部分 120 是否被破坏。在有些实施例中，控制器 190 可以使用硬件芯片组监控计算系统 102 的引导、初始化和/或固件执行的进度。尤其，控制器 190 可以监控引导或初始化期间部分 120 的执行以确定部分 120 是否被破坏或挂起。有些情况下，控制器 190 可以通过监控处理器 140 和固件 110 之间的芯片组而监控处理器 140 的进度。当计算系统 102 引导时，处理器 140 可以访问固件 110 中的缺省地址(例如部分 120 中的一个地址)并根据引导是否成功(例如，通过监控处理器 140 在引导期间的预期响应或通过处理器 140 的处理)将数据返回给固件 110。参考图 1，依照实施例，在计算系统 102 中可以包括芯片组 125 以允许控制器 190 监控固件 110 和处理器 140 之间的通信，以使对该缺省地址的访问和返回数据通过(go through)芯片组 125。如果在引导期间出现了错误，处理器就发送一个错误消息(例如“FF”)作为返回数据给固件 110。同样，来自处理器 140 的错误或其它消息(例如，来自处理器的要与预期响应值比较的响应)可以向控制器 190 表明内核被破坏、固件 110 或其中的一部分被破坏、或者固件或处理器被挂起。特别地，控制器 190 在引导期间或引导之后可以监控芯片组 125 以检测由处理器返回的表示成功引导的有效

数据；或者检测表明引导失败、“内核破坏”、固件破坏或处理器挂起的错误消息。此外，控制器 190 可以使用计算系统 102 的客户机硬件(例如一个芯片组)、控制器 190 的固件或两者以监控客户机系统进度。

在块 215，确定固件是否被破坏或挂起。例如，在块 215，控制器 190 可以确定部分 120 是否被破坏，例如被擦除、有“bug”或者固件的功能或过程中有错误。判定块 215 可以包括根据上述针对块 210 的监控说明进行确定。例如，块 215 可以对应于控制器 190 监控客户机芯片组(例如，图 1 中的芯片组 125)以根据部分 120 或其中的配置数据将处理器 140、操作系统、启动、初始化的进度和它们的预期响应进行比较。如果预期响应与芯片组中检测到的进度不一致，部分 120 可以被看作是已被破坏。

此外，块 215 可以包括监控计算系统 102 的启动或初始化过程是否被挂起。特别地，如果控制器 190 通过计算系统 102 的硬件判定部分 120 被挂起(例如，在计算设备 102 的引导或重置中部分 120 的执行期间)，那么部分 120 可以被看作是被挂起。

如果在块 215 控制器 190 判定部分 120 被破坏或挂起，处理继续到块 220。在块 220，控制器 190 可以使用逻辑、硬件和/或软件机制暂停、停止或暂停客户机系统处理器(例如，使该处理器被“暂停”)。这里，控制器 190 可以使系统处理器(例如处理器 140、客户机系统的系统处理器或中央处理单元 CPU)中止固件、软件或操作系统指令的处理。依照实施例，控制器 190 还可以使计算系统 102 的操作系统停止、挂起、或进入软关机状态。

如果在块 215 控制器 190 判定部分 120 没有被破坏或挂起，处理继续到块 225。在块 225，固件和操作系统被允许执行。例如，块 225 可以对应于控制器 190 允许部分 120 继续或完成在计算系统 102 的启动或初始化期间的执行。在块 225 之后，计算设备 102 可以被允许根据操作系统运行计算过程，例如通过将控制转让给操作系统。另外，依照实施例，在块 225 之后，计算设备 102 可以被允许运行固件 110 的部分 130(以在将控制转让给操作系统之前完成固件 110 的运行)。例如，如图 2 中所示，在“A”，过程 200 可以如图 3 所示在“A”继续，例如，在图 3 中所示并针对其进行说明的块 310 上继续。

在块 230，控制器访问它的本地数据库以访问它的客户机系统专有

信息，例如计算系统 102 的唯一标识。在块 240，控制器 190 使用系统标识信息在网络上向服务器请求更新数据。例如，控制器 190 可以发送一个包括系统标识的请求更新固件消息或“核心被破坏”消息给服务器 180。特别地，控制器 190 可以通过网络 170 将一个包括客户机系统标识和问题标识(如部分 120 遭破坏的表示和/或对最新数据或固件的请求)的预定消息发送给服务器 180(例如通过网络接口 160、通信链路 162 和 182 以及网络 170 发送给运行在服务器 180 上的控制台应用程序)。

在块 245 确定是否存在更新数据。例如，服务器 180 可以从控制器 190 接收对最新固件数据的请求或者“核心遭破坏”消息，并检索服务器 180 中的数据库以确定控制器 190 发送消息中的系统标识所标识系统的固件映像是否存在。更特别地，在块 245，服务器 180 或者在它的本地数据库中查找固件映像，或者发送消息给 IT 管理员以获取所标识的客户机系统(如计算系统 102)的固件映像。如果服务器 180 在特定时间内找到了所标识的客户机的固件映像，它可以使用诸如安全通信、加密数据和/或网络分组在网络 170 上将一个“成功”状态消息和该固件映像发送给计算系统 102。另一方面，如果服务器 180 没有为所标识的客户机系统定位合适的固件映像，服务器 180 用以上为“成功”状态消息所述的类似机制将一个“失败”状态消息发送给计算系统 102。在一种情况下，服务器 180 可以超时查找所标识的客户机系统的合适固件映像的所选搜索时间(例如，如果服务器 180 的数据库中没有该客户机系统标识或其固件)。另一种情况，服务器 180 可以超时等待 IT 管理员的响应，以为所标识的客户机系统提供合适的固件映像。

另外，在块 245，控制器可以轮询硬件以确定是否已经从服务器接收到响应。例如，控制器 190 可以通过网络 170 轮询计算系统 102 的硬件(如网络接口 160)以确定服务器 180 是否已经提供了对固件更新数据请求的响应。在轮询期间，如果在特定的“超时”时间段内没有提供响应，处理可以继续到块 248。

此时，运行在计算系统 102 中的控制器 190 监控网络接口 160 或服务器 180 的响应并依据响应采取适当动作。在块 245，如果不存在更新数据或者轮询超时，处理继续到块 248，在那里向用户通知系统固件错误。例如，如果控制器 190 从服务器 180 接收到一个失败状态消息，控制器 190 可以使用户可理解的消息向用户宣告用户计算系统 102 处在不

可恢复的系统故障错误状态。例如，控制器可以使与计算系统 102 相连的监视器显示一个表示错误状态的消息，或者使计算系统 102 输出表示有错误存在的“滴滴”序列，或者将错误通知给用户。

或者，在块 245，如果更新数据存在并且在轮询超时之前被发送给控制器，轮询处理继续到块 250。在块 250，更新数据被通过网络返回给控制器。这里，如上面为块 245 所述，服务器 180 将一个“成功”状态消息和固件映像发送给计算系统 102，诸如使用网络 170 上的安全通信、加密数据和/或网络分组。

在块 260，更新数据被控制器通过网络从服务器接收到。例如，控制器 190 可能通过网络 170 从服务器 180 接收到一个“成功”状态消息和固件更新数据。固件更新数据可以包括初始固件、固件更新数据或部分 120 的固件更新映像。

接着，在块 270，用该更新数据更新固件。例如，如果控制器 190 从服务器 180 接收到一个成功的状态消息，控制器 190 可以用随该成功状态消息接收到的固件更新数据编程或覆盖部分 120。

在块 280，固件 110 被恢复，控制器 190 消除操作系统、计算系统 102 或处理器 140 上的任何停止或暂停。因而，在任何停止或暂停被消除之后，可以重置系统(如计算系统 102)。例如，块 280 可以返回到过程 200 的块 202 以用更新后的、无破坏的、合适的部分 120 重新引导计算设备 102。

此外，依照实施例，部分 120 可以监控部分 130 以确定部分 130 是否被破坏。部分 120 监控部分 130 的过程可以独立发生或在不包括控制器 190 监控部分 120 的系统中。但是，在这样的系统中，控制器 190 仍然需要向服务器 180 请求并从其接收部分 130 的固件更新数据。相反，部分 120 监控部分 130 的过程可以是包括控制器 190 监控部分 120 的上述说明的实施例，例如在那里在控制器 190 监控部分 120 之后部分 120 监控部分 130。

例如，图 3 是通过网络更新固件的过程 300 的流程图。在块 302，系统被引导或重置。例如，块 302 可以对应于上面对块 202 的说明。接着，在块 303，控制器被启动或初始化，独立于系统继续引导、初始化、运行固件以及运行操作系统或处理的能力。例如，块 303 可以对应于上面为块 203 所做的说明。在块 305，控制器(例如控制器 190)被通知客户

机系统的启动，例如计算系统 102 的启动。例如，块 305 可以对应于上面为块 205 所做的说明。

在块 310，监控系统以确定固件是否遭到破坏。例如，部分 120 可以包括内置到部分 120 的固件中的方法或逻辑以监控、确定或发现部分 130 是否遭到破坏。

如上所述，还预期过程 300 的块 310 可以发生在图 2 中所示块 225 之后。因而，在过程 200 如图 2 中所示使用部分 120 提供了合适的引导之后，可以从块 310 执行过程 300 以确定部分 130 是否被破坏。

例如，在块 310，部分 120 可以监控处理器 140、运行在计算系统 102 上的操作系统、上述客户机芯片组(如图 1 的芯片组 125)或其它硬件以确定部分 130 是否被破坏。此外，在块 310，部分 120 可以执行基于校验和的检验、基于签名的检验和/或基于硬件的检验以确定部分 130 是否被破坏。块 310 还可以包括如上面为块 210 所述那样由控制器 190 监控芯片组 125 和/或监控计算系统 102 的操作系统以确定或检测部分 130 是否被破坏。

在块 315，确定固件是否被破坏或挂起。例如，在块 315，部分 120 可以判定部分 130 是否被破坏，例如被擦除、有“bug”或固件的功能或过程中有错误。判定块 315 可以包括根据上述关于块 310 的监控说明进行确定。例如，块 315 可以对应于部分 120 或控制器 190 监控客户机芯片组(如图 1 中的芯片组 125)，以根据部分 130 或其中的配置数据将处理器 140、操作系统、启动、初始化和它们的期望响应进行比较。如果期望响应与芯片组中检测到的进度(例如，通过包括错误或“FF”消息)一致，部分 130 可以被看作是已遭破坏。

此外，块 315 可以包括确定处理器 140 是否被暂停或停止、确定计算系统 102 的启动或初始化过程是否被挂起、确定运行在计算系统 102 上的操作系统是否被停止、挂起或软关机、和/或确定计算系统 102 的固件(如固件 110)是否被破坏或挂起。例如，块 315 可以对应于部分 120 执行或监控对部分 130 的基于校验和的检验、基于签名的检验或基于硬件的检验。

如果在块 315 判定部分 130 没有被破坏，处理继续到块 325。在块 325，固件和操作系统被允许执行。例如，块 325 可以对应于部分 120 和控制器 190 允许部分 130 继续或完成计算系统 120 的启动或初始化期

间的执行。因而，块 225 可以对应于计算设备 102 被允许根据操作系统运行计算过程，例如将控制转让给操作系统。另外，依照实施例，在块 225 之后，计算设备 102 可以被允许检查自动更新(例如安排在特定时间上的或定时的更新)以更新固件 110、部分 120 和/或部分 130。例如，在“B”，如图 3 所示，过程 200 可以如图 4 所示在“B”上继续(例如，在如图 4 所示并关于图 4 说明的块 415 上继续)。

或者，在块 315，如果判定部分 130 被破坏，处理继续到块 318。在块 318，初始化与控制器通信的硬件。例如，如果部分 120 判定部分 130 被破坏，部分 120 可以使用硬编码的系统参数初始化配置控制器 190 并与之通信所必须的最小硬件集合。部分 120 随后发送一个预定消息给控制器 190，带有标识部分 130 被破坏的参数。同样，部分 120 可以发送一个预定消息到控制器 190，带有请求部分 120 和/或 130 的更新或恢复的参数。部分 120 然后在上述由硬编码的系统参数初始化的硬件上轮询来自控制器 190 的响应。

在块 318 之后，过程 300 继续到块 330，在那里客户机系统标识被访问。块 330 可以对应于控制器 190 访问客户机系统(如计算系统 102)的标识。例如，块 330 可以对应于上述块 230。

在块 330 之后，处理继续到块 340，在那里通过网络请求更新固件数据。例如，控制器 190 可以在网络 170 上用上面为块 240 所述的消息和系统标识信息，向服务器 180 请求部分 120 和/或 130 的更新数据。另外，在块 340 发送的系统标识信息可以包括为固件 110 的部分 120 和/或 130 请求的固件的标识。

在块 340 之后，处理继续到块 345，在那里确定更新数据是否存在。块 345 可以对应于上述块 245。例如，服务器 180 可能没有找到与所提供的系统标识对应的合适的更新固件数据，或者控制器 190 轮询硬件超时。

同时，运行在计算系统 102 中的控制器 190 监控为响应网络接口 160 或服务器 180，并根据该响应采取适当动作。在判定块 345，如果更新不存在，处理继续到块 348，在那里向用户通知系统固件错误，如为块 248 所述。另外，在块 348，控制器 190 向部分 120 发送表示没有获得部分 130 的固件的错误响应。因此，部分 120 可以接收到自控制器 190 发出的失败状态消息，并通知用户，例如通过显示器、滴滴声或其它计算

系统 102 可用的方法。

在块 345，如果更新数据存在，处理继续到块 350，在那里更新数据被通过网络返回给控制器。块 350 可以对应于以上对块 250 的描述。处理然后在块 360 上继续，在那里控制器通过网络从服务器接收到更新数据。

在块 360，控制器通过网络从服务器接收到更新数据。例如，控制器 190 可以通过网络 170 从服务器 180 接收到一个“成功”状态消息和固件更新数据。该固件更新数据可以包括原始固件、固件更新数据、部分 120 和/或 130 的固件更新映像。特别地，控制器 190 可以从服务器 180 下载部分 120 和/或 130 的更新数据，并且将该更新数据复制到系统存储器(如存储器 142)中并发送一个成功状态消息给部分 120。或者，控制器 190 可以准备一个控制台或服务器直通接口以将更新数据直接下载到部分 120 和/或 130(例如通过用更新数据直接覆盖部分 120 和/或 130 中的当前数据)并发送成功状态消息给部分 120。

接着，在块 370，用该更新数据更新固件。例如，如果控制器 190 从服务器 180 接收到一个成功状态消息，控制器 190 可以用和该成功状态消息一起接收到的固件更新数据编程或覆盖部分 120 和/或 130。此外，如果部分 120 接收到一个成功状态消息，部分 120 可以读取存储器(如存储器 142)中存储的更新数据或者可以从上述直通接口考虑或读取数据，以将更新数据编程、写入或覆盖到部分 120 和/或 130 中。

在块 380，固件 110 被恢复并且控制器 190 和/或部分 120 重置客户机系统。例如，部分 120 可以用系统专有的过程重置计算系统 102。例如，块 380 可以返回过程 300 的块 302 以用最新的、可靠的、合适的部分 130 重新引导计算设备 102。

因而，认为块 230、240、245、248、250、260 和 270 可以只应用于访问、请求、查询、返回、接收和更新部分 130，例如由部分 130 控制在这些块中所采取的写或更新部分 130 的动作。或者，那些块可以应用于部分 120 和部分 130，例如由控制器 190 在相同动作集中控制那些动作来写或更新部分 120 和 130。

另外，依照实施例，固件 110 可以被自动更新，例如通过检验并从服务器 180 接收更新而被在特定的时间或定时地更新(如更新部分 120 和/或 130)。自动更新固件 110 的过程可以独立发生或发生在不包括控制

器 190 监控部分 120 和/或部分 120 监控部分 130 的系统中。但是，在这样的系统中，控制器 190 仍然需要向服务器 180 请求并从其接收固件 110 的更新数据。相反，具有自动更新固件 110 的过程可以是包括控制器 190 监控部分 120 和/或部分 120 监控部分 130 的上述说明的实施例。

例如，图 4 是通过网络自动更新固件的过程 400 的流程图。在块 402，系统被引导或重置。例如，块 402 可以对应于对块 202 的上述说明。接着，在块 403，控制器被启动或初始化，独立于系统的能力以继续引导、初始化、运行固件、运行操作系统或处理。例如，块 403 可以对应于对块 203 的上述说明。在块 405，控制器(如控制器 190)被通知客户机系统起动，例如计算系统 102 的启动。例如，块 405 可以对应于对块 205 的上述说明。

在块 415，监控系统以判定是否是自动更新固件 110 的全部或一部分的时间。依照实施例，控制器 190 或部分 120 可以包括监控时钟信号以确定是否是所选时间或自从上次自动更新或尝试已经过去的所选时间段的逻辑。因而，块 415 可能出现而不管计算系统 102 的引导、处理器 140、操作系统被或不被停止、阻塞、挂起或软关机。如果所选时间或时间段已经过去，现在就是自动更新部分 120 和/或部分 130 的时间。

如上所述，还预期过程 400 的块 415 可以发生在图 3 所示块 325 之后。因而，在过程 300 如图 3 中所示用部分 130 提供了适当的引导之后，可以执行从块 415 开始的过程 400，以确定是否需要自动更新。

在块 415，如果不是自动更新固件的时间，处理继续到块 425。在块 425 固件和操作系统被允许执行。例如，块 425 可以对应于固件 110 和控制器 190 允许计算系统 102 的启动或初始化继续。块 425 还可以对应于计算设备 102 的操作系统正在运行时，因而可以允许计算设备 102 继续根据操作系统运行计算过程。

或者，在块 415，如果是自动更新固件的时间，处理继续到块 418。在块 418，初始化与控制器通信的硬件。例如，如果是自动更新的时间，部分 120 可以使用硬编码的系统参数初始化配置控制器 190 并与之通信所必需的最小硬件集合。部分 120 随后发送一个预定消息给控制器 190，带有标识到了自动更新部分 120 和/或部分 130 的时间的参数。

或者，部分 120 可以发送一个表示部分 130 被破坏和/或请求更新或恢复部分 130 的消息给控制器 190。这种情况下，块 418 可以对应于对

块 318 的上述说明。部分 120 然后在如上所述由硬编码的系统参数初始化的硬件上轮询来自控制器 190 的响应。例如，部分 120 可以监控网络接口 160 或控制器 190，例如通过(经由网络接口 160)轮询计算系统 102 的硬件以确定服务器 180 是否已经响应。

在块 418 之后，过程 400 继续到块 430，在那里访问客户机系统标识。块 430 可以对应于控制器 190 访问客户机系统的标识(如计算系统 102)和/或要自动更新的固件。例如，块 430 可以对应于上述块 230 或块 330。此外，在块 430，如果到了自动更新的时间，控制器 190 可以访问它的本地数据库以查找客户机系统标识以及固件 110 的修正信息、发布日期等。

在块 440，控制器 190 使用系统标识信息通过网络向服务器请求更新数据。例如，控制器 190 可以发送一个包括系统标识的自动更新消息、请求固件更新消息或“核心遭破坏”消息给服务器 180。

特别地，控制器 190 可以向服务器 180 发送一个包括客户机系统标识以及修正信息、发布日期(例如部分 120 和/或部分 130 的修正信息、发布日期等)等的预定消息。从控制器 190 到服务器 180 的消息的传递可以对应于对块 240 的说明。

因此，在块 445，服务器 180 将从控制器 190 接收到的消息解释为自动更新请求，并在服务器 180 的数据库中查找比来自控制器 190 的消息中提供的修正信息、发布日期等更新或更晚的修正。或者，在块 445，服务器 180 可以发送一个消息给 IT 管理员以请求该客户机系统的映像。

同时，运行在计算系统 102 中的控制器 190 为响应监控网络接口 160 或服务器 180 并根据该响应采取适当的动作。检索数据库、轮询并确定更新数据在服务器 180 是否存在可以对应于对块 345 的说明。例如，在判定块 445，如果部分 120 的轮询判定控制器 190 在选择的超时时间段之前没有接收到来自服务器 180 的消息，处理继续到块 448，在那里固件 110 向用户发送一个表示“网络不可用”(例如网络 170)的消息，例如通过从计算系统 102 所显示的消息。

另外，在判定块 445，如果更新不存在，例如服务器在所选超时时间段之前没有在服务器上找到匹配，服务器 180 向控制器 190 发送一个表示“无自动更新可用”的消息。然后，过程 400 继续到块 448，在那里处理器 190 使用户被通知“无自动更新可用”，例如通过从计算系统

102 显示的消息。

此外，在块 448，控制器 190 可以发送“无自动更新可用”消息给固件 110。在固件 110 接收到“无自动更新可用”消息后，固件 110 可以使用户获得“网络不可用”消息的通知，例如通过从计算系统 102 显示的消息。在块 448，在固件 110 接收到“无自动更新可用”或“网络不可用”消息后，固件 110 可以允许计算系统 102 继续引导、启动、初始化或运行操作系统，因为过程 400 是用于自动更新(例如假定固件没有被破坏或挂起)。

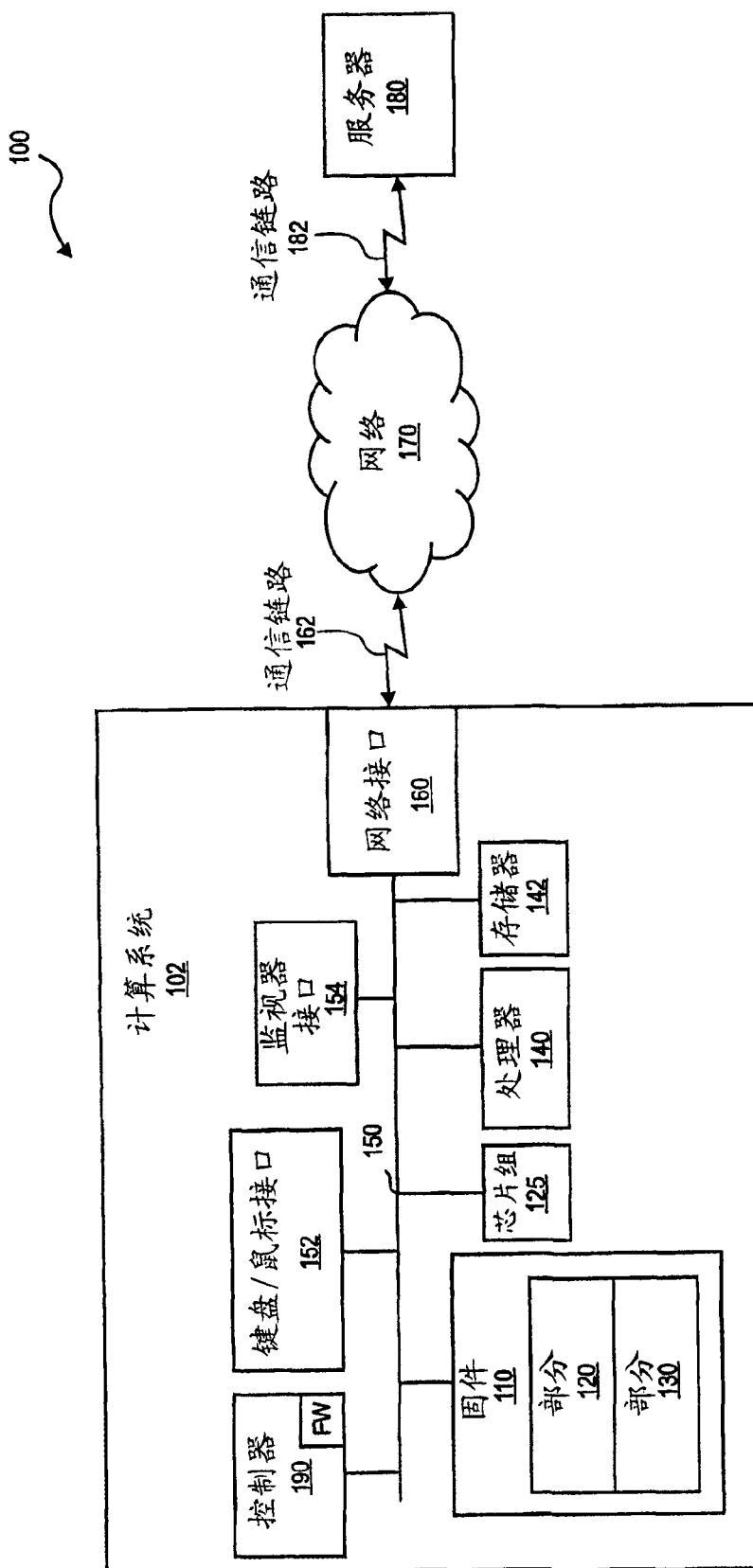
在块 445，如果更新数据存在，处理继续到块 450，在那里更新数据被通过网络返回控制器。块 450 可以对应于前面对块 250 的说明。例如，在块 450，服务器 180 可以使用诸如安全服务器会话、加密数据和/或网络分组返回一个“成功”状态消息以及计算系统 102 的固件映像的适当后来的修正。处理随后继续到块 460，在那里控制器通过网络从服务器接收到更新数据。

在块 460，控制器通过网络从服务器接收到更新数据。例如，控制器 190 可能通过网络 170 从服务器 180 接收到一个“成功”状态消息和固件更新数据。该固件更新数据可以包含初始固件、固件更新数据、或固件 110 的全部或部分的固件更新映像。特别地，控制器 190 可以从服务器 180 下载固件 110 的更新数据并将该更新数据复制到系统存储器(如存储器 142)中并发送一个成功状态消息给部分 120。或者，控制器 190 可以准备一个控制台或服务器直通接口以将该更新数据直接下载到固件 110(如通过用更新数据覆盖固件 110 中的当前数据)并发送一个成功状态消息给部分 120。

接着，在块 470，用该更新数据更新固件。例如，如果控制器 190 从服务器 180 接收到一个成功状态消息，控制器 190 可以用和该成功状态消息一起接收到的固件更新数据编程或覆盖固件 110(如部分 120 和部分 130)。此外，如果部分 120 接收到一个成功状态消息，部分 120 可以读取存储器(如存储器 142)中存储的更新固件数据或者从上述直通接口考虑或读取更新数据以用该更新固件数据编程、写或覆盖固件 110。这里，轮询硬件以获取响应的部分 120，将从系统存储器(如存储器 142)或通过服务器直通接口接收到该成功状态消息和更新或更晚修正的固件映像，以写或允许更新部分 130。

在块 480，更新固件 110 并且控制器 190 和/或部分固件 110 重置客户机系统。例如，部分 120 可以用系统专有的过程重置计算系统 102。还预期块 480 可以返回过程 400 的块 402，以用自动更新后的固件 110 重新引导计算设备 102。

在前面的说明书中，描述了具体的实施例。但是，在不偏离如权利要求中所阐述的实施例的广义的精神和范围的前提下，可以进行各种改进和变化。因此，说明书和附图将被看作是说明性而非限制性的。



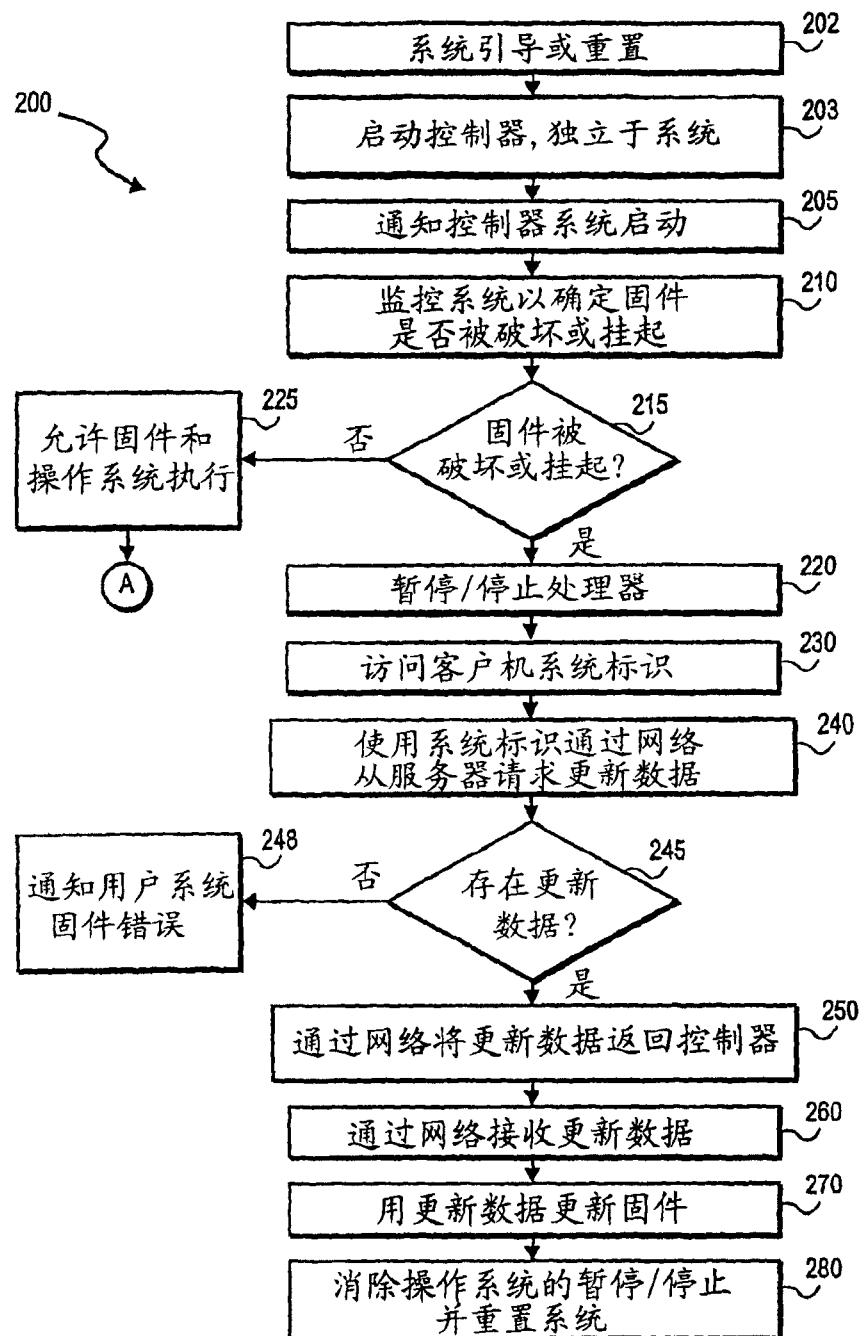


图 2

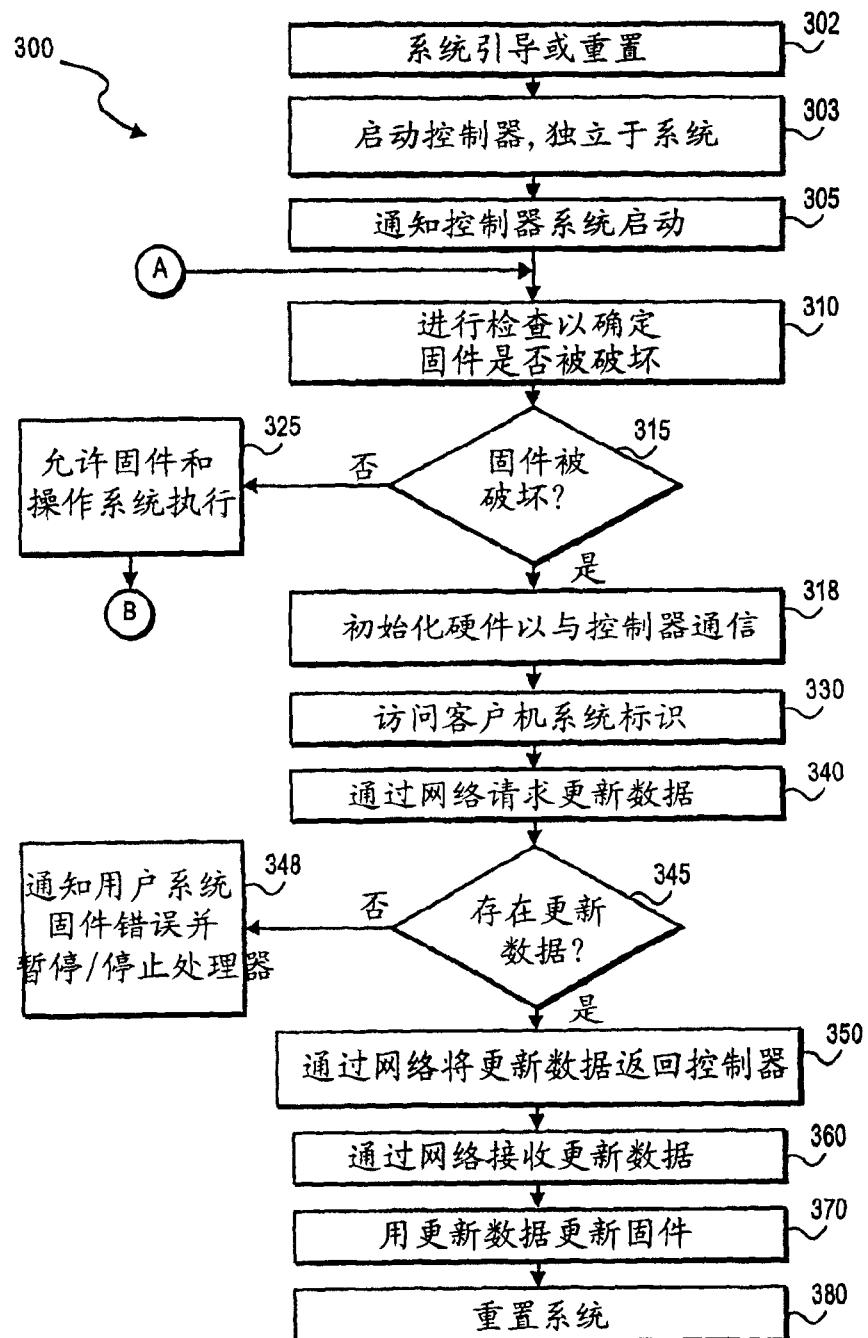


图 3

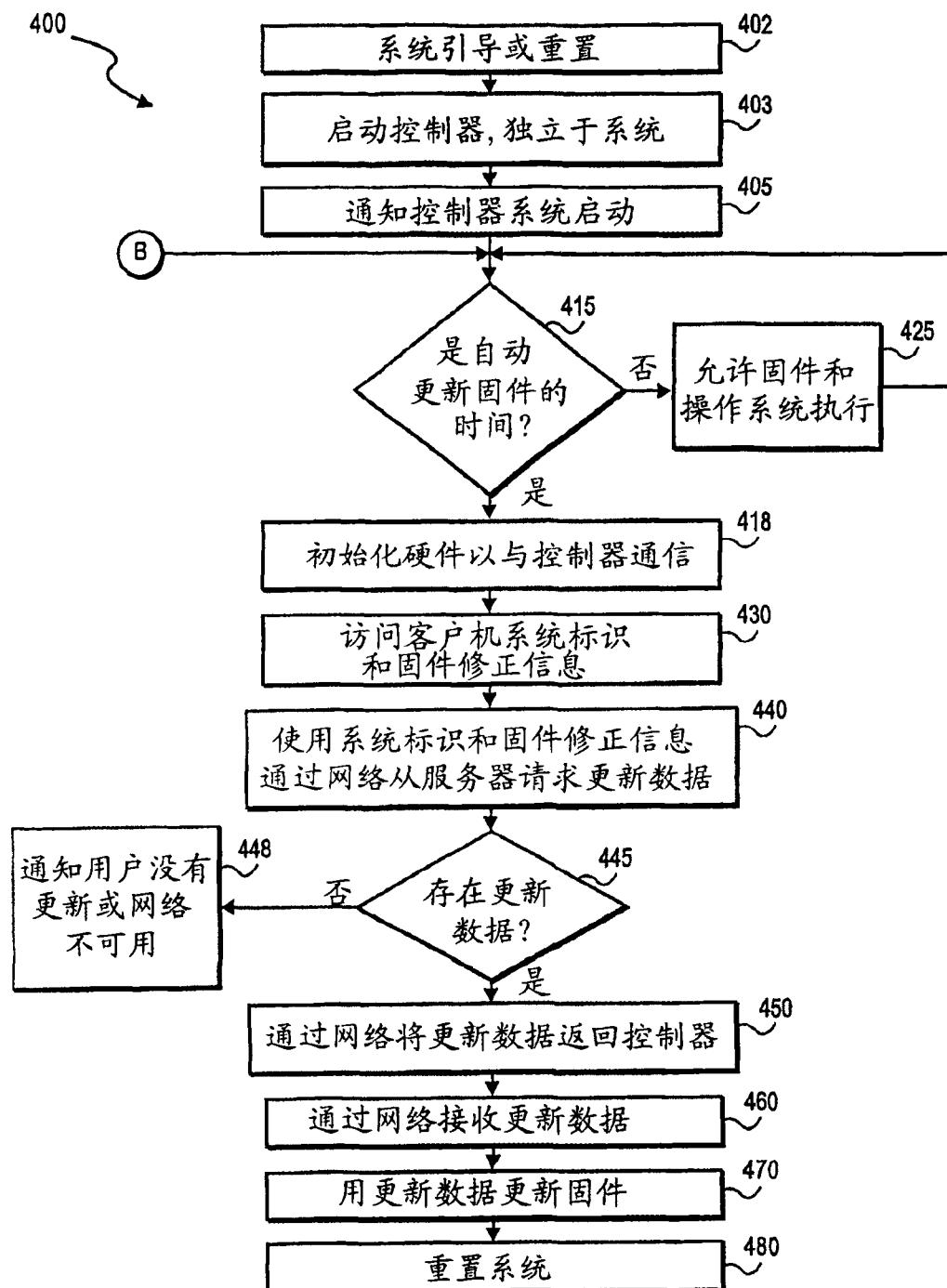


图 4