[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 3/00 (2006.01)
G06F 9/44 (2006.01)
G06F 15/16 (2006.01)
H04L 9/00 (2006.01)

[12] 发明专利说明书

专利号 ZL 200480001332.6

[45] 授权公告日 2008 年 9 月 3 日

[11] 授权公告号 CN 100416465C

[22] 申请日 2004.7.23
[21] 申请号 200480001332.6
[30] 优先权
[32] 2003.12.15 [33] US [31] 10/737,708
[86] 国际申请 PCT/US2004/023975 2004.7.23
[87] 国际公布 WO2005/060388 英 2005.7.7
[85] 进入国家阶段日期 2005.5.20
[73] 专利权人 微软公司
地址 美国华盛顿州
[72] 发明人 M·E·默伦曼斯 A·埃弗布奇
J·罗伯特 K·肖曼
M·默哈姆德 J·G·达德兹
[56] 参考文献
US2002/0029227A1 2002.3.7
CN1409239A 2003.4.9
US5845077A 1998.12.1
US6199204B1 2001.3.6
WO01/90892A1 2001.11.29
US6202207B1 2001.3.13
CN1310388A 2001.8.29
审查员 李 俊
[74] 专利代理机构 上海专利商标事务所有限公司
代理人 张政权

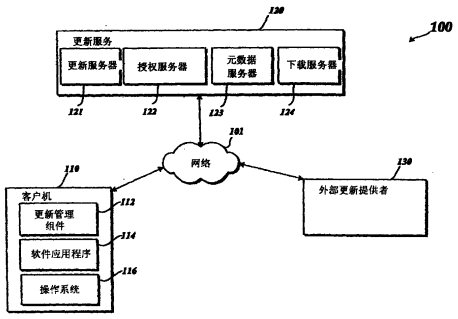
权利要求书 3 页 说明书 24 页 附图 14 页

[54] 发明名称

管理和传递软件更新的系统和方法

[57] 摘要

一种便于软件更新的选择和实现，同时最小化选择和实现软件更新所需的带宽和处理资源的系统和方法。 在一个实施例中，更新服务控制对储存在服务器上的软件更新或其它类型的软件的访问。



1．一种在包括与软件更新服务通信的客户机计算装置的计算机系统中用于将软件更新从所述软件更新服务传递到所述客户机计算装置的方法，其特征在于，所述方法包括：

（a）获取从所述客户机计算装置到所述软件更新服务的同步请求，其中，如果所述客户机计算装置储存了已安装软件更新，则所述同步请求包括所述已安装软件更新的标识符；

（b）确定所述同步请求是否包括至少一个已安装软件更新的标识符；

（c）如果确定所述同步请求包括至少一个已安装软件更新的标识符，则选择一额外的软件更新以传递到所述客户机计算装置，其中，对所述额外的软件更新的选择取决于至少实现所述额外的软件更新中定义的第一先决条件，所述第一先决条件要求所述同步请求包括至少一个已安装软件更新的标识符；

（d）如果确定所述同步请求不包括至少一个已安装软件更新的标识符，则选择第一级软件更新以传递到所述客户机计算装置，其中，所述第一级软件更新不包括一先决条件；

（e）将所选择的软件更新的指令部分从所述软件更新服务传递到所述客户机计算装置；

（f）如果所述客户机计算装置包含实现储存在所选择的软件更新的指令部分中的一适用性规则的条件的至少一个组件，则将所选择的软件更新的指令部分储存在所述客户机计算装置中，作为已安装软件更新；

（g）如果所述客户机计算装置不包含实现所选择的软件更新中储存的一适用性规则的至少一个条件的至少一个组件，则在所述客户机计算装置中储存所选择的软件更新的指令部分，作为不成功的软件更新；

（h）确定所述软件更新服务不包含对所述客户机计算装置可用的额外的软件更新；以及

（i）将所选择的软件更新的本地化数据部分从所述软件更新服务传递到所述客户机计算装置。

2．如权利要求 1 所述的方法，其特征在于，

确定所述软件更新服务不包含对所述客户机计算装置可用的额外的软件更新

包括：确定所选择的软件更新的至少一个指令部分提供了所述软件更新服务包含对所述客户机计算装置可用的额外软件更新的指示；以及

当确定所选择的软件更新的至少一个指令部分提供了所述软件更新服务包含对所述客户机计算装置可用的额外软件更新的指示时，重复（a）－（f）。

3．如权利要求 2 所述的方法，其特征在于，所述方法还包括：

生成包含从所传递的本地化数据部分中获得的文本的显示，其中，所述文本提供了对所选择的软件更新的描述。

4．如权利要求 3 所述的方法，其特征在于，所述方法还包括：

获取标识与所选择的软件更新相关联的一选择的数据文件的命令；以及

响应于接收所述命令，将与所选择的软件更新相关联的数据文件从所述软件更新服务传递到所述客户机计算装置。

5．如权利要求 1 所述的方法，其特征在于，所述方法还包括：

确定所述客户机计算装置是否储存了所述已安装软件更新；

如果确定所述客户机计算装置储存了所述已安装软件更新，则在所述同步请求中提供所述已安装软件更新的标识符。

6．如权利要求 1 所述的方法，其特征在于，所述先决条件定义了一条件，该条件要求所述同步请求中所涉及的软件更新的组合，其中，所述条件是由连接所述软件更新的组合的至少一个逻辑操作符定义的。

7．如权利要求 1 所述的方法，其特征在于，所述方法还包括：

如果所述客户机计算装置储存了所述不成功的软件更新，则获取所述同步请求中所述不成功的软件更新的标识符；

如果确定所述同步请求包括所述不成功的软件更新的标识符，则限制与所述不成功的软件更新有关的所选择的软件更新的传递。

8．如权利要求 1 所述的方法，其特征在于，所述同步请求由在所述客户机计算装置上执行的自动更新程序模块生成。

9．如权利要求 1 所述的方法，其特征在于，所述同步请求由请求手动更新的用户调用。

10．如权利要求 1 所述的方法，其特征在于，所选择的软件更新被配置成更新安装在所述客户机计算装置上的一个或多个文件，其中，将所选择的软件更新从所述软件更新服务传递到所述客户机计算装置包括：

获取标识用于更新已安装文件的至少一个版本的多个可用增量补丁的自提取

文件；

获取安装在所述客户机计算装置上的一个或多个文件的详细目录；

选择一个或多个可适用的增量补丁来实现所选择的软件更新，其中，选择所述一个或多个可适用的增量补丁对应于将标识所述多个可用增量补丁的所述自提取文件映射到安装在所述客户机计算装置上的所述一个或多个文件的详细目录；以及

发送对一个或多个所选择的增量补丁的请求。

11．一种在包括与软件更新服务通信的客户机计算装置的计算机系统中用于将软件更新从所述软件更新服务传递到所述客户机计算装置的方法，其特征在于，所述方法包括：

（a）在所述软件更新服务处获取来自所述客户机计算装置的同步请求，所述同步请求包括标识储存在所述客户机计算装置上的每个已安装软件更新的第一部分、以及标识储存在所述客户机计算装置上的不成功的软件更新的第二部分；

（b）基于所述同步请求的所述第一部分确定可用更新的列表；

（c）根据所述同步请求的所述第二部分过滤所述可用更新的列表，以获得选择的软件更新的列表；

（d）将每个选择的软件更新的指令部分从所述软件更新服务传递到所述客户机计算装置，所述指令部分标识了所述选择的软件更新的任何先决条件；

（e）在所述软件更新服务处接收来自所述客户机计算装置对至少第一个选择的软件更新的本地化数据部分的请求；以及

（f）将所述第一个选择的软件更新的本地化数据部分从所述软件更新服务传递到所述客户机计算装置，所述本地化数据部分包括描述所述第一个选择的软件更新的通用信息。

管理和传递软件更新的系统和方法

发明领域

本发明涉及软件和计算机网络，尤其涉及管理和传递软件更新的系统和方法。

发明背景

市场上最常见的软件产品要经受连续的修订过程，以修补或升级特征和/或功能。软件产品或组件的每一次修订可能需要添加新的文件和/或用更加新版本的文件替换现有的文件。一旦销售商隔离了软件产品问题并创建了该问题的解决方案，他可能希望将该修补放入更新中，并令更新对顾客广泛地可用。软件销售商具有尽可能快和无故障地向顾客分发软件更新的商业动机。

因特网为顾客提供了一种获得软件产品的最近更新的重要渠道。因特网使用率的蓬勃发展造成了顾客对软件产品和更新在线提供用于下载的常见期望。促进因特网的使用来分发更新也是软件销售商所感兴趣的，因为它减少了他们的成本，并允许顾客在修补可用于下载后立即可获得所确定问题的修补。因特网上的销售商站点可被设计成令发现和查找应用程序的更新文件是非常简单的。文件下载的技术方面大多数已从用户的观点中消失，并且它现在通常由操作系统来处理。

在常规的方法中，软件销售商将软件更新构造为“包”用于下载。包通常是自提取的可执行文件，它具有设置程序，并且产品的更新文件的每一个被嵌入并被压缩以使包更小。包的大小通常是每一改变的文件的压缩后大小的总和，加上提取代码本身的大小。在执行时，包将所包含的每一文件提取到一临时位置，然后开始设置程序以将每一文件安装到系统目录中一正确的位置。以压缩形式发货的文件在它们被安装时被解压。同一位置中名字相同的任一现有文件将简单地被替换文件所覆盖。

即使因特网令软件更新的广泛和快速分发成为可能，然而网络传输的有限带宽导致了问题。常见软件应用程序的绝对大小导致了更新的下载大小变得不合理的大。通常，用于产品的各种问题的多个修补将被组合成一个更新。如果销售商以常规的基础更新软件产品，则更新包的下载大小将继续增长，这是因为销售商无法在

假设用户已经具有来自较早更新的那些文件的假设上省略这些文件。由于更新包组合了若干的完整文件，即使文件被压缩，它也可能相当大。有时，即使在最快的调制解调器连接上，下载的带宽效率也被降低。

常规下载过程的耗时方面当然是不合需要的。在某些情况下，顾客在下载这些文件的过程中支付了长距离或连接时间的费用。连接时间的任何减少将减少这些顾客的直接金钱成本。销售商通常也具有关于他们所提供的下载大小的可辨识成本，因此减小尺寸也可给予他们直接的金钱收益。减小下载的尺寸将增加他们的可用网络带宽，从而允许他们用现有的网络服务器设备来服务更多的顾客。

下载大的更新所花费的长时间也令下载过程变得更易受各种网络连接问题的攻击。对于为什么因特网话路可能会被过早地断开连接有多个原因，包括电话线噪声、呼叫等待信号以及无意的命令。某些因特网服务供应商实施了连接时间限制，限制了用户可以在单个话路中在线的时间量。如果当网络连接被切断时用户正在下载大文件，则他或她可能必须从头开始。最常见的操作系统和文件传输协议不允许文件传输的再继续，因此任何中间的进度将会丢失，并且传输必须重新开始。失败的几率如此之大，以致于许多用户发现在线获取更新几乎是不可能的。如果更新包的尺寸太大，则用户可能永远也无法完整地下载它。

减小软件更新的尺寸并提高带宽效率的一种尝试涉及增量补丁，或二进制补丁的使用。本领域的技术人员可以理解，增量补丁对应于当由计算装置执行时修改现有文件的专用软件代码。由于增量补丁包括专用软件代码，因此对文件的每一唯一版本需要唯一的增量补丁。当应用于软件更新时，软件更新服务可发送较小尺寸的更新增量补丁，而非发送完整的更新文件。已更新的增量补丁然后用于将现有文件修改成已更新的文件。

尽管更新增量补丁可潜在地减少更新文件所需的数据量，然而现有的增量打补丁方法在存在大量的文件版本的情况下管理可适用的增量文件的选择时是不够的。由于对文件的每一版本需要唯一的增量补丁，因此典型的软件更新系统通常会要求上百（如果不是上千）个对应于文件的每一唯一版本的唯一增量补丁。在一种方法中，支持增量打补丁的某些更新服务将所有可能的增量补丁发送到客户机计算装置。然而，这一方法通常在可能更新增量补丁的数量增加时也增加了实现软件更新所需的数据量。因此，潜在可适用的增量补丁的数量可迅速增长为与完整的更新文件相同的大小。在另一种方法中，网络化更新软件服务扫描客户机机器以选择对每一客户机机器可应用哪一增量补丁。尽管这减少了所发送的增量补丁信息量，然

而它需要软件更新服务上的附加逻辑来扫描客户机机器并选择可适用的增量补丁。附加逻辑的使用增加了必须由服务提供的系统资源。此外，该方法通常阻止如通常由传统的 web 服务器所实现的网络高速缓存的使用。

除上述缺点之外，现有的系统无法传送某些类型的软件更新，如硬件驱动程序。如本领域中所已知的，专用软件更新，如应用于硬件驱动程序的软件更新，难以在大规模分发的基础上提供给用户，因为大多数专用软件更新仅在具有特殊硬件的客户计算机上起作用。例如，在大多数情况下，如果客户计算机获得了不兼容的硬件驱动程序升级，则该驱动程序升级的安装会导致致命的错误，或甚至会阻止计算机运行。

如可以从上述内容容易地理解的，需要一种具有服务器和多个客户机之间的软件更新的改进通信的系统和方法。另外，对具有允许更新服务在传送专用更新时以特定类型的客户机为目标的改进机制的软件更新系统和方法有需求。

发明概述

本发明针对的是用于管理软件更新的系统和方法。更具体地，本发明针对的是便于软件更新的选择和实现，同时最小化选择和实现软件更新所需的带宽和处理资源的系统和方法。依照本发明的一方面，提供了一种用于处理软件更新的系统和组件体系结构。

依照本发明的一方面，提供了一种用于将软件更新从软件更新服务传递到客户机计算装置的方法。该方法可以在包括与软件更新服务通信的客户机计算装置的计算机系统中实现。依照该方法，软件更新服务从客户机计算装置获取同步请求。如果客户机计算装置储存了已安装的软件更新，则该同步请求包括已安装软件更新的标识符。软件更新服务确定该同步请求是否包括至少一个已安装软件更新的标识符。如果软件更新服务确定同步请求包括至少一个已安装软件更新的标识符，则软件更新服务选择一额外的软件更新以传递到客户机计算装置。对额外软件更新的选择取决于额外软件更新中定义的先决条件的实现，其中，该先决条件需要同步请求包括至少一个已安装软件更新的标识符。或者，如果软件更新服务确定同步请求不包括至少一个已安装软件更新的标识符，则软件更新标识符选择第一级软件更新以传递到客户机计算装置，其中，第一级软件更新不包括先决条件。软件更新服务然后将所选择软件更新的指令部分从客户机计算装置传递到软件更新服务。然后，如果客户机计算装置包含实现所选择软件更新中储存的可适用性规则的条件的至少

一个组件，则软件更新服务将所选择软件更新的指令部分储存在客户机计算装置中，作为已安装的软件更新。

依照本发明的另一方面，提供了一种用于将软件更新从软件更新服务传递到客户机计算装置的方法。该方法可以在包括与软件更新服务通信的客户机计算装置中实现。依照该方法，软件更新服务从客户机计算装置获取授权请求。该授权请求包括储存在客户机计算装置上的客户机授权模块的标识符。软件更新服务确定客户机计算装置是否与一目标组相关联。如果客户机授权模块指示客户机计算装置与目标组相关联，则软件更新服务确定该客户机计算装置与该目标组相关联。如果确定客户机计算装置与目标组相关联，则软件更新服务将服务器 cookie 从软件更新服务传递到客户机计算装置，其中服务器 cookie 标识了该目标组。软件更新服务获取对储存在软件更新服务上的至少一个软件更新的请求，其中该请求包含服务器 cookie。响应于获取请求，软件更新服务确定软件更新是否与服务器 cookie 中标识的目标组相关联。另外，如果软件更新服务器确定软件更新与服务器 cookie 中标识的目标组相关联，则软件更新服务将软件更新从软件更新服务传递到客户机计算装置。

依照本发明的又一方面，提供了一种用于将数据结构从服务器传递到客户机计算装置的方法。该方法可以在包括与服务器通信的客户机计算装置的计算机系统中实现。依照该方法，如果客户机计算装置包含指示客户机计算装置与一目标组相关联的授权模块，则软件更新服务确定客户机计算装置是否与该目标组相关联。如果确定客户机计算装置与目标组相关联，则软件更新服务将服务器 cookie 从服务器传递到客户机计算装置。服务器 cookie 标识了该目标组。软件更新服务获取对储存在服务器上的至少一个数据结构的请求。该请求包含服务器 cookie。响应于获取请求，软件更新服务确定数据结构是否与服务器 cookie 中标识的目标组相关联。如果服务器更新服务确定该数据结构与服务器 cookie 中标识的目标组相关联，则软件更新服务将该数据结构从服务器传递到客户机计算装置。

附图的简要描述

当结合附图阅读以下详细描述，可以更好地理解并更容易地明白本发明的上述方面以及许多附加的优点，附图中：

图 1 是依照本发明的软件更新系统的框图，包括客户机计算机以及提供更新软件的更新服务；

图 2 是图 1 的软件更新系统的框图，它依照本发明示出了向更新服务认证客户机计算装置；

图 3 是图 1 的软件更新系统的框图，它依照本发明示出了客户机计算装置和更新服务之间的可用更新的同步；

图 4 是图 1 的软件更新系统的框图，它依照本发明示出了软件更新信息从更新服务到客户机计算装置的发送；

图 5 是图 1 的软件更新系统的框图，它依照本发明示出了客户机计算装置对更新信息的处理和选择；

图 6 是图 1 的软件更新系统的框图，它依照本发明示出了客户机计算装置对增量补丁的合并和对更新文件的安装；

图 7 是依照本发明的软件更新例程的流程图，它示出了由客户机计算装置和更新服务实现的用于识别可用于安装在客户机计算装置上的软件更新；

图 8 是依照本发明提供对储存在更新服务上的更新的选择性访问的授权例程的协议图；

图 9 所示是依照本发明的授权例程的一组示例软件更新的框图；

图 10 是依照本发明将选择的一组软件更新从软件更新服务传递到客户机计算装置的同步例程的协议图；

图 11 所示是依照本发明用于显示对各个客户机计算装置可用的软件更新列表的图形用户界面的示例性部分的框图；

图 12A 和 12B 所示是依照本发明由客户机计算装置 110 实现以检索和安装请求的软件的软件更新处理子例程 1200；以及

图 13 所示是依照本发明由客户机计算装置实现的用于更新基线安装组件的子例程的流程图。

具体实施方式

一般而言，本发明针对的是管理软件更新的系统和方法。更具体地，本发明针对的是便于软件更新的选择和实现，同时最小化选择和实现软件更新所需的带宽和处理资源的系统和方法。依照本发明，软件更新可对应于用于特定软件应用程序或操作系统的更新。此外，软件更新可包括软件驱动程序或对固件的更新，如系统 BIOS。依照本发明的一个方面，提供了一种处理软件更新的系统和组件体系结构。依照本发明的另一方面，提供了一种便于向更新服务授权和同步客户机机器的更新

协议和接口。依照本发明的又一方面，提供了一种适用增量补丁更新安装组件和各种已安装文件的方法。然而，本领域的技术人员可以理解，本申请中也可提供本发明的其它方面。此外，相关领域的技术人员可以理解，每一所标识的方面可被个别或作为公共的发明性方面的一部分来考虑。

图 1 的软件更新系统 100 所示是依照本发明的软件更新系统 100 的框图。一般而言，软件更新系统 100 可包括一个或多个客户机计算装置 110、更新服务 120 以及外部更新提供者 130。一般而言，更新服务 120 储存并管理传递到并安装在客户机计算装置 110 上的软件更新的分发。软件更新可由更新服务 120 或任意数量的外部更新提供者 130 来提供。

客户机计算装置 110、更新服务 120 和外部更新提供者 130 通过网络 101 电子地通信。网络可以是局域网（LAN）或更大的网络，如广域网（WAN）或因特网。通过使用一般已知的软件，软件更新系统 100 可被配置成在客户机计算装置 110 和更新服务 120 的服务器 121、122、123 和 124 之间交换文档、命令和其它已知类型的信息。本领域和其他领域的技术人员可以理解，图 1 所示的软件更新系统 100 是用于实现本发明的一个合适的系统的简化示例，本发明并非局限于此示例。

如后文更详细地描述的，更新服务 120 的一个实施例包括若干服务器。如图 1 所示，更新服务 120 包括更新服务器 121，用于管理更新服务 120 的全过程并协调更新服务 120 的服务器 121、122、123 和 124 的处理。授权服务器 122 生成客户机请求的授权 cookie，这些授权 cookie 进而用于生成允许客户机计算机访问更新服务 120 提供的更新的服务器 cookie。元数据服务器 123 提供关于更新服务 120 提供的更新的一般信息。元数据服务器 123 允许本发明的系统识别用于特定类型的客户机计算机或特定的客户机计算机组的特定更新。下载服务器 124 提供了用于传送与由更新服务 120 提供的软件更新相关联的数据文件的一个或多个软件组件。

外部更新提供者 130 可包括分发软件更新的一个或多个服务器。外部更新提供者 130 可与提供软件、软件更新或要分发到客户机计算机组的其他数据的实体相关联。例如，外部更新服务器 130 可以与期望使用更新服务 120 来分发一个或多个软件应用程序的更新的第三方软件开发者相关联。在另一示例中，外部更新提供者 130 可以与软件更新系统 120 相关联。

客户机计算装置 110 可以是储存并执行软件应用程序 114 的任何计算装置。客户机计算装置 110 可以从多个不同的计算机产品的任一个形成，包括但不限于，个人计算机（PC）、个人数字助理（PDA）、移动电话、双向寻呼机等等。如本

领域或其他领域的普通技术人员所理解的，客户机计算装置 110 的体系结构可以采用任一合适的形式。例如，客户机计算装置 110 可包括用于提供与网络 101 的通信的网络接口。网络接口可以被配置成用于任何有线或无线网络连接，并用于任一合适的通信协议，如 TCP/IP 协议。另外客户机计算装置 110 可包括处理单元、显示器和存储器单元。存储器单元可储存操作客户机计算装置 110 所必须的程序代码，如操作系统 116。另外，存储器单元储存用于控制和执行本发明的过程的更新管理组件 112。

软件更新系统 100 储存软件程序，当被执行时，它们实现本发明。当被执行时，软件更新系统 100 储存、管理并选择性地传递软件更新。如下文更完整地描述的，除其它益处之外，本发明提供了用于定义和选择有资格接收软件更新的客户机计算装置的目标组的机制。本发明也提供了用于下载与软件更新相关联的数据文件的改进的机制。

为说明本发明的目的，提供了本发明的工作示例的详细描述。在描述该工作示例时，参考软件更新，它指的是软件应用程序的特定升级，例如，将媒体播放器版本 6.0 升级到媒体播放器版本 7.0。如本领域的普通技术人员所理解的，这一软件更新可包括与软件更新相关联的多个数据文件的传递和安装。由此，为说明本发明的目的，在软件更新和包含软件更新的个别数据文件之间作出区分。

现在参考图 2-6，描述更新客户机计算装置 110 上的一个或多个文件的软件更新系统 100 的组件之间的说明性交互。参考图 2，软件更新服务通过由一个或多个外部更新提供者 130 发送软件更新信息来启动。如上所述，外部更新提供者 130 可以与软件更新系统 100 相关联。或者，软件更新信息可以由第三方外部更新提供者 130 发送。在本发明的一个说明性实施例中，软件更新信息可包括用于更新文件的软件代码、用于替换文件的软件代码、用于确定软件更新的可适用性的各种规则、和/或描述软件更新的显示信息。软件更新信息的发送可在任一时刻完成，并且不必要与其它所示的软件更新组件交互同时发生。

在从外部更新提供者 130 接收了软件更新信息之后，更新服务 120 生成一段或多段数据，以便于发送更新信息。数据可包括对应于用于更新文件的不同版本的软件增量补丁集的补丁存储文件。数据也可包括对应于将特定的文件版本映射到补丁存储文件中找到的对应增量的索引的补丁存储清单。数据还可包括对应于更新代理将用于请求和安装特定软件更新数据的信息的自提取文件，如后文更详细地描述的。相关领域的技术人员可以理解，补丁存储文件、补丁存储清单和自提取文件的

生成可以在任一时刻完成，并不必要与其它所示的组件交互同时发生。

为启动对客户机的软件更新信息发送，客户机计算装置 110 启动对更新服务 120 的认证请求。在本发明的一个说明性实施例中，认证请求对应于客户机计算装置 110 和更新服务 120 之间的更新协议交互，这将在后文更详细描述。在完成了认证之后，更新服务 120 将认证 cookie 发送到客户机计算装置 120。现在参考图 3，已认证的客户机计算装置 120 然后启动与更新服务器 120 的可用更新的同步。在本发明的一个说明性实施例中，同步请求也对应于客户机计算装置 110 和更新服务 120 之间的更新协议交互，这将在后文更详细地描述。在完成了同步之后，客户机计算装置 110 接收所有可适用的软件更新的信息和描述更新的信息。然而，在本发明的一个说明性实施例中，没有下载任何启动更新的软件代码。

继续参考图 3，在更新过程期间的某一时刻，接收到要安装的更新的选择。在本发明的一个说明性实施例中，可向用户呈现在同步期间接收到的软件更新信息，并要求选择一个适当的更新。或者，客户机计算装置 110 可以配备自动选择所有可适用的软件更新的方式。此外，客户机计算装置 110 也可具有允许它自动选择可用软件更新的一个子集的某些规则。再者，用户可通过诸如经由互联网网页与更新服务 120 通信来启动更新的选择。

现在参考图 4，如果尚不存在更新代理，则更新管理组件 112 例示客户机计算装置 110 上的更新代理 118。更新代理 118 然后请求诸如自提取文件等软件更新信息包的发送。更新代理 118 接收自提取文件并执行对安装程序的任何更新，如后文更详细地描述的。此外，更新代理 118 可向更新服务 120 请求任何丢失或被破坏的信息。

现在参考图 5，一旦更新代理 118 接收到了软件更新信息包，更新代理 118 执行安装在客户机计算装置 110 上的文件的清查。基于清查和软件更新信息包的比较，更新代理 118 确定需要哪一增量补丁或其它更新信息来完成所选择的更新。更新代理 118 然后发送对特定增量更新的请求。在本发明的一个实施例中，对软件更新的请求可对应于通过直接网络连接发送的直接请求，被称为手动更新。在本发明的另一实施例中，对软件更新的请求可以是不需要明显的用户行动而发送的后台请求。这一实施例被称为自动更新。

在本发明的一个说明性实施例中，如果软件更新对应于增量补丁，则更新代理 118 向更新服务 120 发送标识由包存储清单标识的特定增量补丁的请求。或者，在增量补丁不可用或者若干增量补丁失败的情况下，更新代理 118 可启动一后退过

程。后退过程可包括对来自包存储文件的整个更新文件的完整副本的发送的请求。后退过程也可包括对自主式包内的整个更新文件的完整副本的发送的请求。

在本发明的一个说明性实施例中，更新服务 120 的下载服务器 124 可直接处理来自更新代理 118 的软件更新请求。或者，请求也可由任意数量的附加外部下载服务器来处理，如从更新服务 120 接收所请求的更新增量补丁的传统 web 服务器。例如，企业可使用内部服务器来更新客户机机器。另外，请求可由外部下载服务器来处理，其中，在处理先前的请求时高速缓存某些或所有更新增量补丁。另外，在此实施例中，下载可以被分发到能够服务超文本传输协议（“HTTP”）数据请求的若干另外的下载服务器。

参考图 6，一旦接收到了软件更新信息，更新代理 118 将增量补丁与已安装文件合并，以生成更新文件。另外，更新代理 118 可确认合并程序是否成功地更新了适当的文件。如上所述，如果增量补丁不能被确认，则更新代理 118 可再次请求增量补丁或在若干次失败之后请求整个更新文件。一旦更新代理 118 获取了已被确认和更新文件，该文件被安装在客户机计算装置 110 上。

图 7 依照本发明的软件更新处理例程 700 的流程图，它示出了客户机计算装置 110 和软件更新服务 120 之间的交互。在框 702，软件更新服务 120 授予对客户机计算机 110 的访问权限。在本发明的一个说明性实施例中，授予对客户机计算机的访问权限的过程可包括生成允许访问与特定的计算机组相关联的软件更新的服务器签发的 cookie。授权过程的更详细解释将参考图 8 来描述。

在框 704，客户机计算机 110 和软件更新服务 120 同步更新信息。在本发明的一个说明性实施例中，软件更新服务 120 向客户机计算装置 110 发送描述特定软件更新的元数据。元数据包含描述可用软件更新的信息，以允许用户选择一个或多个更新来安装。同步过程的更详细描述将在下文参考图 9 和 10 来描述。在框 706，客户机计算装置 110 获取对要下载的可适用更新的选择。在本发明的一个说明性实施例中，对可适用更新的选择可对应于适用多个唯一用户界面以便于用户选择。用户界面的选择将在下文参考图 11 来更详细描述。

在框 708，客户机计算装置 110 处理对可适用软件更新的用户选择，并与软件更新服务 120 接口以请求特定的更新信息。在本发明的一个说明性实施例中，客户机计算装置 110 选择并请求一个或多个可适用更新增量补丁。客户机计算装置 110 上的更新代理 118 然后可处理所请求的数据以实现所选择的软件更新。在框 710，例程 700 终止。

参考图 8，现在描述授予对客户机计算装置 110 的访问权限并对应于框 702（图 7）的协议图 800。在本发明的一个说明性实施例中，软件更新服务 120 使用可扩充定目标机制，以控制客户机计算机 110 对更新和其它软件的访问。软件更新服务 120 结合了将特定软件更新与客户机计算装置 110 的一个或多个目标组相关联的机制。例如，软件更新服务 120 可限制特定的硬件驱动程序更新对具有特定硬件设备的特定品牌的客户机计算装置 110 的访问。在这一示例中，软件更新服务 120 可定义具有特定品牌名和特定硬件设备的客户机计算装置 110 的目标组，并限制特定软件下载向该目标组的发送。

在本发明的一个说明性实施例中，可扩充定目标机制可通过使用定义客户机计算装置到一个或多个目标组的成员资格的软件组件（“授权插件”）来促进。授权插件在客户机计算装置 110 上的存在定义了客户机计算装置是否属于授权插件的特定目标组。例如，目标组可包括具有特定软件应用程序的有效产品标识（“PID”）号的所有计算机。在这一示例中，如下文参考图 8 更详细地描述的，授权插件 826 可安装在客户机中，以从客户机计算装置的存储器模块中读取 PID，并将所获取的 PID 传递到对应的 PID 服务器插件 829。对应的 PID 插件此处也被称为 PID 确认程序 829，使用了一种或多种方法来确定接收到的 PID 是否有效。一旦确定了储存在客户机计算装置 110 上的 PID 有效，则服务器生成指示客户机计算装置 110 是具有有效 PID 的目标组的成员的服务器 cookie。在另一示例中，目标组可包括被指定为 β 测试（测试第二版）计算机的客户机计算装置。

在本发明的一个说明性实施例中，软件更新服务 120 的授权服务器 122 包含多个服务器授权插件，它们定义了授权服务器将识别的一组客户机计算装置目标组。每一服务器授权插件包含与储存在客户机计算装置 110 上的对应客户机授权插件传递数据的组件。以类似的方式，每一客户机计算装置 110 包括标识客户机所属的目标组的一个或多个授权插件。在本发明的一个说明性实施例中，客户机授权插件可在安装或升级软件应用程序期间，如安装或升级操作系统期间，被安装在每一客户机计算装置中。另外，服务器授权插件可以由期望控制对软件更新的访问的管理员动态地安装或移除。储存在客户机计算装置 110 和授权服务器 122 上的授权插件可以是实际的软件插件，或者授权插件可以被硬编码（hard code）成动态链接库。

如图 8 所示，授权服务器 122 包含三个示例服务器授权插件：（1）第一服务器授权插件 828，它定义了包括所有计算机的目标组（后文称为“所有计算机目标组”）；（2）第二服务器授权插件 829，它定义了包括具有有效 PID 的计算机的

目标组（后文称为“PID 目标组”）；以及（3）第三服务器授权插件 830，它定义了包括β测试计算机的目标组（后文称为“β测试目标组”）。同样如图 8 所示，客户机计算装置 110 包含两个客户机授权插件：（1）第一客户机授权插件 825，它指示客户机计算装置 110 是“所有计算机目标组”的成员；以及（2）第二客户机授权插件 826，它指示客户机装置 110 是“PID 目标组”的成员。在此示例中，客户机计算装置 110 不包含指示它是“β测试目标组”的成员的授权插件。如本领域的普通技术人员所理解的，每一客户机授权插件 825 和 826 可被配置成执行客户机计算装置 110 上的一个或多个功能，以协助确认过程。例如，第二客户机授权插件 826 可被配置成检查客户机计算装置 110 的存储器，以验证或获得已安装软件应用程序的 PID。

如图 8 所示，授权子例程 702 在客户机计算装置 110 将配置请求 803 传递到授权服务器 122 时开始。在本发明的一个说明性实施例中，配置请求 803 是从被配置成获取描述储存在授权服务器 122 上的授权插件的信息的任一合适的软件组件形成的。如本领域的技术人员所理解的，配置请求 803 可使用一种被称为“GetConfig（获取配置）”的已知方法。响应于接收配置请求 803，授权服务器 122 传递配置响应 804，它包括标识储存在授权服务器 122 上的所有授权插件的信息。在一个实施例中，配置响应 804 包括串数组，它标识并描述了储存在授权服务器 122 上的所有授权插件。在本示例中，配置响应 804 包括标识第一服务器授权插件 828、第二服务器授权插件 829 以及第三服务器授权插件 830 的信息。

在框 805，客户机计算装置 110 响应于接收配置响应 804 生成一个或多个授权 cookie。在框 805 的处理中，客户机计算装置 110 对每一对匹配的客户机和服务器授权插件生成授权 cookie。由此，在本示例中，第一客户机授权插件 825 生成与“所有计算机目标组”相关联的第一授权 cookie，因为第一客户机授权插件 825 和第一服务器授权插件 828 都与“所有计算机目标组”相关联。另外，第二客户机授权插件 826 生成与“PID 目标组”相关联的第二授权 cookie，因为第二客户机授权插件 826 和第二服务器授权插件 829 都与“PID 目标组”相关联。未生成第三授权 cookie，因为客户机计算装置 110 不具有指示它是“β测试目标组”的成员的授权插件。

如本领域的技术人员也可以理解的，框 805 的过程的一个实现可包括对本领域中称为“GetAuthCookie（获取授权 cookie）”的一般已知的软件方法的使用。也可以理解，每一授权 cookie 的生成可涉及额外的处理。例如，第二客户机授权插件 826 可以被配置成检查储存在客户机系统注册表中的信息，以检索 PID 并在

授权 cookie 中包括该 PID。在其它示例中，框 805 的过程可包括与其它计算机或装置通信的过程。例如，客户机授权插件可与诸如声卡、扫描仪、视频卡等设备通信，以获取设备的构造和模型。在其它非限制示例中，客户机授权插件可与诸如指纹读取器等安全设备通信，以获取描述用户的信息。

一般而言，客户机授权插件可从客户机计算装置 110 的任何组件或通信上耦合至客户机计算装置 110 的任何其它计算装置读取配置信息。在其它示例中，客户机授权插件可被配置成使用一个或多个公钥或私钥应用编程接口（API），以收集并加密来自客户机的信息，这些信息将由对应的服务器插件来确认。在这些示例中，PID 确认器插件 826 使用私钥 API 来加密客户机的 PID，以将经加密的 PID 传递到服务器以供解密和确认。在其它实施例中，其它客户机授权插件可使用诸如指纹读取器或语音打印等生物测定措施，以构造传递到服务器供确认的授权 cookie。在又一示例中，客户机授权插件可调用 web 服务或任一其它服务以将授权凭证或任一其它类型的数据传递到授权服务器 122。

在本发明的一个说明性实施例中，每一授权 cookie 包括标识相关联的目标组的串。例如，串可指示特定的授权 cookie 与“PID 目标组”相关联。每一授权 cookie 也包括用于在客户机和服务器之间传递数据的数据段。例如，与“PID 目标组”相关联的授权 cookie 可具有包含实际 PID 的数据段。如本领域的普通技术人员所理解的，数据段可包含以诸如字节数组等任一格式储存的任何类型的数据。例如，如果客户机和服务器上的插件需要公钥和私钥的传递，则这类数据可在一个或多个授权 cookie 的数据段中加密。

一旦客户机计算装置 110 对每一对对应的客户机和服务器授权插件生成了授权 cookie，客户机计算装置将所生成的授权 cookie 传递到授权服务器 122。如图 8 所示，客户机计算装置 110 在 cookie 请求 806 中传递授权 cookie。cookie 请求 806 包括用于传递框 805 的处理中生成的授权 cookie 的数组的任一合适的格式。这一部分授权方法 702 的一种实现可包括对本领域中被称为“GetCookie（获取 cookie）”的一般已知的软件方法的使用。

在一个实施例中，cookie 请求 806 也包括储存在客户机计算装置 110 的存储器中的其它授权服务器 cookie。如可以从以下描述中更容易地理解的，客户机计算装置 110 的存储器可储存在授权例程 700 的先前的执行中创建的旧授权服务器 cookie。通过在 cookie 请求 806 中提供储存的授权服务器 cookie，客户机计算装置 110 能够维护其在授权子例程 702 的先前的执行中被授予的访问特权。在本示例中，

由于没有授权服务器 cookie 储存在客户机中，因此 cookie 请求 806 包括与“所有计算机目标组”相关联的第一授权 cookie 以及与“PID 目标组”相关联的第二授权 cookie。

下一步，如框 807 中所示，响应于接收 cookie 请求 806，授权服务器 122 生成服务器 cookie。在一个实施例中，对于每一接收到的授权 cookie，向适当的服务器授权插件作出调用以生成服务器 cookie 数据。每一服务器授权插件生成的服务器 cookie 数据包括接收到的授权 cookie 中标识的每一目标组的标识符。在本示例中，由于 cookie 请求 806 包括与“所有计算机目标组”相关联的第一授权 cookie，以及与“PID 目标组”相关联的第二授权 cookie，因此授权服务器 122 生成包括这些相应目标组的标识符的服务器 cookie 数据。在授权服务器 122 上，如果在 cookie 请求 806 中接收到旧服务器 cookie，则将服务器 cookie 数据与旧服务器 cookie 组合，以生成新服务器 cookie。在一个实施例中，新服务器 cookie 通过使用诸如三重 DES 等公众可用的加密方法来加密。

在本发明的一个说明性实施例中，服务器 cookie 可包括标识一个或多个相关联的目标组的经加密的信息。另外，服务器 coookie 可包括过期数据，它以明文格式和加密格式两者储存。以明文格式储存的过期数据由客户机计算装置 110 用于监视服务器 cookie 的过期。以加密格式储存的过期数据由软件更新服务 120 用于确定客户机计算装置 110 是否被授权来接收与特定目标组相关联的更新。在一个实施例中，服务器 cookie 的过期数据应用于服务器 cookie 中标识的所有目标组。或者，或除应用于整个服务器 cookie 的过期时间之外，服务器 cookie 可包括多个过期数据，其每一个可应用于个别的目标组。如本领域的普通技术人员所理解的，每一服务器 cookie 可包括附加数据。例如，服务器 cookie 可被配置成储存客户机状态信息，如授权子例程 702 的最后执行的时间标记。

一旦被生成，授权服务器 cookie 809 从授权服务器 122 传递到客户机计算装置 110。下一步，如框 811 所示，服务器 cookie 然后被储存在客户机计算装置 110 的存储器中。当客户机计算装置 110 确定服务器 cookie 的至少一个组件过期时，客户机计算装置可重新执行授权方法 702 来获取新的服务器 cookie。如上所述，在授权方法 702 的每一次随后的执行中，客户机计算装置 110 可在 cookie 请求 806 中将其储存的服务器 cookie 传递到授权服务器 122。在一个实施例中，客户机不必发送请求 803，除非服务器通知客户机，服务器配置已改变，即，添加了新的授权插件。

依照本发明的另一方面，软件更新服务 120 可提供用于在元数据服务器 123 和客户机计算装置 110 之间同步更新信息的同步子例程。通过使用唯一的软件更新层次，同步子例程可有效地标识应用于特定客户机计算装置的特定更新。另外，通过使用授权子例程 702 中生成的服务器 cookie，同步子例程可选择性地授予对与特定目标组相关联的更新的访问权限。

依照本发明的一个说明性实施例，每一软件更新包括三个部分 1）指令部分；（2）本地化的数据部分；以及（3）数据部分。如本领域的普通技术人员所理解的，每一更新可具有上述部分的一个或多个。例如，更新可包含指令部分、本地化的数据部分和数据流部分。在另一示例中，更新可仅包含用于测试客户机计算装置的一个或多个条件的指令部分。软件更新的各个部分在下文更详细地描述。

一般而言，指令部分包含两个子部分：（1）定义要由客户机计算装置 110 测试的一个或多个条件的适用性规则；以及（2）标识个别更新的正确安装所需要的一个或多个更新的一组先决条件。如后文所描述的，适用性规则可定义与计算机相关的多个条件，这些条件的每一个可通过使用任何逻辑操作符与其它条件相关联。例如，指令部分可包括确定计算机是否安装了 Windows®的特定版本的适用性规则。同样如下所述，该组先决条件可标识要求先前已被安装的一个或多个更新。例如，如后文参考图 9 更详细地描述的，个别的更新可包含列出个别更新的正确安装所需要的其它更新的先决条件。在其它示例中，同样如图 9 所示，该组先决条件可包括对逻辑操作符的使用以定义更复杂的先决条件规则。

指令部分也包含指示是否由依赖于特定更新的其它更新的代码，如布尔标志。为说明的目的，如果没有依赖于特定更新的其它更新，则该更新被认为是 LEAF 更新。用于指示更新是否为 LEAF 的布尔标志是在当添加或移除相关更新时由元数据服务器 123 动态地更新的。

每一更新的本地化数据部分包括描述更新的通用信息。例如，本地化数据部分可包括描述更新的特征和益处的信息。本地化数据部分也可包括更新的安装过程的文本描述。另外，本地化数据部分可包括与更新有关的任何其它数据或信息。例如，本地化数据可指示更新是高优先级更新。在另一示例中，本地化数据可提供特殊的安装消息，如指示更新不能与其它软件更新一起安装的消息。本地化信息可以是允许向用户显示其包含的信息的格式。

每一更新的数据部分包括更新的一个或多个二进制数据流。在一个实施例中，每一更新的数据部分可以与一个或多个数据文件相关联，如可执行文件、文档、链

接库等等。如下文更详细地描述的，每一更新可以与数据文件的组合相关联，其每一个促进了客户机使用的软件的实际升级、安装或修改。例如，更新的安装可以通过使用包括完成所选择的更新所需的所有信息的单个 CAB 文件来促进。或者，更新的安装可通过使用用于更新储存在客户机计算装置上的一个或多个文件的多个个别更新来促进。

在本发明的一个说明性实施例中，软件更新可以按允许软件更新的受控分发的分层结构来排列。一般而言，更新的层次定义了更新之间的关系，并特别地指示了哪些更新依赖于其它更新。为说明的目的，图 9 中提供并示出了一组示例性更新。如图所示，示例更新 900 的层次包括基础更新组 901、第二更新组 902 以及第三更新组 903。一般而言，基础更新集 901 中的每一更新不具有需要其它更新的安装的先决条件。然而，第六更新 921 包含需要第一更新 911、第二更新 912、第三更新 913 的安装的先决条件。第七更新 922 包含需要第四更新 914 的安装的先决条件。第八更新 931 包含需要第六更新 921 和第五更新 915 的先决条件。因此，第八更新 931 也需要第一更新 911、第二更新 912 和第三更新 913 的安装。为说明本发明的目的，给定示例更新集 900 的所有更新都与“所有计算机目标组”和“PID 目标组”相关联。

如图 9 所示，并且如后文更详细描述的，每一更新包含指定更新安装的条件的适用性规则。例如，第一更新 911 需要操作系统的英语版本的安装。第二更新 912 需要 Windows® XP 版本 SP1 的安装。在另一示例中，第六更新 921 需要被称为 XP PATCH1 的软件补丁的安装。因此，如果未满足适用性规则，则客户机计算装置 110 将不安装更新。

图 9 也示出了指示是否有依赖于特定更新的其它更新的指令部分段。该段被称为 LEAF，它可以是指示软件更新是一系列相关更新的最后一个更新的布尔值形式。为说明本发明的目的，如果没有其它更新列出特定更新为先决条件，则该特定更新是 LEAF 更新。如图 9 所示，第七更新 922 和第八更新 931 是仅有的两个 LEAF 更新。

图 10 示出了依照本发明形成的同步子例程 704（图 7）的协议图。一般而言，同步子例程 704 选择性地在客户机计算装置 110 和诸如元数据服务器 123 等服务器之间传递某些更新的指令部分，以标识可应用于客户机计算装置 110 的更新。如图 10 所示，为启动更新，客户机计算装置 110 首先处理已安装的更新，并将同步请求 1051 传递到元数据服务器 123，请求对客户机可用的一个或多个更新。响应于

接收同步请求 1051，元数据服务器 123 向客户机计算装置 110 返回多个更新。如可以从以下描述中更容易地理解的，客户机计算装置 110 在传递同步请求 1051 之前处理本地储存的数据。

客户机计算装置 110 处理每一接收到的更新的指令部分，以确定是否能满足适用性规则中定义的条件。如果满足了个别的更新中定义的条件，为说明同步子例程 1050 的目的，个别更新是“已安装”的，并且所安装的更新被保存在客户机更新高速缓存的第一组成部分中。另一方面，如果未满足个别更新中定义的条件，则该个别更新被认为“失败”，并且失败的更新被保存在客户机更新高速缓存的第二组成部分中。在同步子例程 1050 的这一描述中，如果更新被安装，则可假定满足了先决条件和适用性规则的条件。为描述该子例程的目的，更新的安装不必要意味着与更新相关联的数据文件实际上被安装到客户机计算装置 110 中。

在一个实施例中，客户机更新高速缓存的两个组成部分可用于归类所接收到的更新。第一组成部分用于储存已安装的、非 LEAF 更新；第二组成部分用于储存客户机接收的所有其它更新，即未安装的更新。更新高速缓存的第二组成部分也可包括所有 LEAF 更新的存储。如下文更详细地描述的，储存在更新高速缓存中的更新可以被传递到元数据服务器 123 并由其处理，以标识可用于在客户机计算装置 110 上的安装的其它相关更新。

现在回头参考图 10，将描述同步请求的细节，被示出为项 1051、1055 和 1060。如本领域的普通技术人员所理解的，同步请求可由多个不同装置、进程、应用程序、用户启动的命令之一来启动，以请求更新。同步请求可由请求更新列表的用户、由客户机代理启动的自动更新、以及向元数据服务器 123 或更新服务 120 请求信息的任何其它软件组件启动。在一个实施例中，同步请求包括授权服务器 cookie，如从授权例程 702 生成的授权服务器 cookie。服务器 cookie 的使用允许服务器确定客户机是否为一个或多个目标组的成员。

每一同步请求也可包括储存在客户机更新高速缓存中的每一更新的标识符。更具体地，如果一个或多个更新储存在更新高速缓存中，同步请求包括具有用于已安装、非 LEAF 更新的标识符的第一部分；以及具有用于诸如 LEAF 更新、已失败更新以及未安装的其它更新等所有其它更新的标识符的第二部分。更新标识符可以是任何格式，包括但不限于，整数数组。或者，如果没有更新储存在客户机更新高速缓存中，则同步请求不配备更新标识符。当同步请求设备配备更新标识符时，同步请求提供客户机计算装置 110 没有任何高速缓存的更新的指示。

如图 10 所示，第一同步请求 1051 从客户机计算装置 110 传递到元数据服务器 123。在本示例中，如果这是方法的第一次执行，则客户机的更新高速缓存将不包含任何更新。由此，第一同步请求 1051 不包含用于高速缓存的更新的标识符。响应于接收同步请求，如框 1052 所示，元数据服务器 123 确定同步请求是否包含至少一个更新标识符。如果确定同步请求不包含更新标识符，则元数据服务器 1123 通过选择用于传递到客户机计算装置 110 第一级更新来响应。如上所述，第一级更新可包括没有标识其它更新的先决条件的任何更新。

或者，如果确定同步请求包含至少一个更新标识符，则元数据服务器 123 检查服务器的已储存更新的先决条件，以选择用于传送到客户机的额外更新。在一个实施例中，元数据服务器 123 选择已实现先决条件的更新。在检查先决条件时，服务器使用同步请求的第一部分的更新，它包括安装在客户机上的非 LEAF 更新的标识符。

除选择已实现先决条件的更新之外，服务器也使用同步请求的第二部分中标识的更新，以过滤所选择的更新。更具体地，同步请求的第二部分中标识的未安装更新、LEAF 更新和已失败更新用于过滤一个或多个所选择的更新。本发明的这一特征允许本发明的系统和方法避免储存在元数据服务器 123 上的更新的多次发送。

返回到本示例，由于第一同步请求 1051 不包括更新标识符，则元数据服务器 123 选择更新 901 的基础级，以传递到客户机计算装置 110。参考图 9 所示的示例更新集，基础集更新 901 包括标号为 911、912、913、914 和 915 的更新。

在框 1052 的处理中，元数据服务器 123 也检查包含在同步请求 1051 中的授权服务器 cookie，以标识与客户机计算装置 110 相关联的目标组。元数据服务器 123 也检查在框 1052 的过程中选择的更新的目标组。框 1052 的处理然后过滤出不与接收到的授权服务器 cookie 中标识的目标组相关联的所有所选择的更新。在本示例中，由于所有所选择的更新 911、912、913、914 和 915 与 PID 和所有计算机目标组相关联，因此所有所选择的更新都被发送到客户机计算装置 110。

元数据服务器 123 然后在同步响应 1053 中将所选择的更新传递到客户机计算装置 110。一般而言，每一同步响应包括服务器 120 发送的每一更新的指令部分。由此，在本示例中，第一同步响应 1053 包括标号为 911、912、913、914 和 915 的更新的指令部分。在一个实施例中，每一同步响应不包括每一更新的本地化数据部分或数据部分。

下一步，如框 1054 中所示，客户机计算装置 110 处理每一接收到的更新的指

令部分，以确定是否可满足适用性规则中定义的条件。再次参考图 9，客户机计算装置 110 处理接收到的更新 911-915 的指令部分。为说明本发明的目的，在本示例中假定客户机计算装置 110 的操作系统是 Windows®版本 XP SP1 的英语安装。也假定客户机计算装置 110 是 Dell PC，并运行 32 位的 X86 处理器。由此，在处理示例更新集的指令部分时，客户机计算装置 110 可确定将满足第一更新 911 中定义的条件，因为计算机包含英语 OS。将满足第二更新 912 中定义的条件，因为操作系统是 Windows®版本 XP SP1。将满足第三更新 913 中定义的条件，因为客户机计算装置运行 X96 处理器。将满足第五更新 915 中定义的条件，因为客户机计算装置 110 是 Dell PC。结果，第一更新 911、第二更新 912、第三更新 913 和第五更新 915 都被保存在客户机更新高速缓存的第一组成部分中。将不满足第四更新 914 中定义的条件，因为客户机计算装置 110 不运行 64 位的 X86 处理器。由此，第四更新 914 被认为是已失败的更新，并保存在客户机更新高速缓存的第二组成部分中。

返回到图 10，在框 1054 的处理中，客户机计算装置 110 也确定是否需要后续的同步请求。在一个实施例中，如果至少一个所接收到的更新指示它不是 LEAF 更新，则确定需要后续的同步请求。在本发明的示例中，确定需要随后的同步请求，因为所有接收到的更新都不是 LEAF 更新。由此，客户机计算装置 110 将后续的同步请求 1055 传递到元数据服务器 123。

如上所述，同步请求包括储存在客户机更新高速缓存中的每一更新的标识符。由此，在本示例中，后续的同步请求 1055 包括指示客户机上安装了第一更新 911、第二更新 922、第三更新 913 和第五更新 915 的第一数据部分。另外，后续的同步请求 1055 包括指示客户机上未成功地安装第四更新 911 的第二数据部分。

响应于接收后续的同步请求 1055，如上所述，元数据服务器 123 确定后续的同步请求 1055 是否包含至少一个更新标识符。如果确定后续的同步请求包含至少一个更新标识符，元数据服务器 123 检查所有储存的更新的先决条件，以选择用于传送到客户机的额外更新。

再次参考本示例，在框 1056 的处理中，元数据服务器 123 将选择第六更新 921，因为满足了它的先决条件。更具体地，如图 9 所示，第六更新 921 被选中用于传递到客户机计算装置 110，因为其需要第一更新 911、第二更新 912 和第三更新 913 的安装的先决条件已被实现。第七更新 922 和第八更新 931 将不被选中用于传递到客户机，因为它们的先决条件未被实现。更具体地，同步请求 1055 不包含第四更新 914 的标识符，它是第七更新 922 的先决条件。另外，同步请求 855 不包含第六

更新 921 的标识符，它是第八更新 931 的先决条件。

返回到图 10，同步子例程 1050 通过在后续响应 1057 中将所选择的更新从元数据服务器 123 传递到客户机计算装置 110 来继续。再次参考本示例，后续响应 1057 将包括与第六更新 921 有关的信息，它在后续响应 1057 中被传递到客户机 110。

在接收到后续响应 1057 之后，客户机计算装置 110 处理后续响应 1057 的指令部分。类似于框 1054 的过程，客户机计算装置 110 处理每一接收到的更新的指令部分，以确定是否满足适用性规则中定义的条件。在本示例中，如果假设在客户机计算装置中已安装了 XP PATCH1，则第六更新 921 被认为是已安装的，并且更新被写入客户机计算装置 110 的更新高速缓存中。由于第六更新 921 不是 LEAF 更新，因此客户机计算装置 110 发送包括储存在客户机更新高速缓存的第一和第二组成部分中的所有更新的另一同步请求 1060。同步请求 1060 也包括授权服务器 cookie。

在本示例中，通过使用上述元数据服务器 123 的处理，同步请求 1060 在框 1061 被处理，其中，服务器选择第八更新 931。第八更新 931 被选中，因为同步请求 1060 指示第五和第六更新 915 和 921 已安装在客户机计算装置 110 上。假设第八更新 931 与授权服务器 cookie 中标识的相同目标组相关联，则在另一响应 1062 中将第八更新 931 的指令部分传递到客户机计算装置 110。然后在框 1063 以类似于框 1054 和 1059 的处理的方式处理第八更新 931。由于响应 1062 所有所接收的更新都是 LEAF 更新，因此不将后续请求发送回元数据服务器 123。

在客户机计算装置 110 处，在确定了所有接收到的更新都是 LEAF 更新之后，或者如果在响应 1062 中没有接收到任何更新，则同步子例程 1050 将驱动程序同步请求 1064 从客户机计算装置 110 传递到元数据服务器 123。如本领域的普通技术人员所理解的，驱动程序同步请求 1064 可包括描述安装在客户机计算装置 110 上的所有硬件的信息以及描述已安装软件的信息。类似于先前的软件同步请求（1051、1055 和 1060），驱动程序同步请求 1064 可将已安装的更新传递到服务器。另外，当前在客户机上高速缓存的所有驱动程序更新（如果有的话）被传递到服务器。

响应于接收驱动程序同步请求 1064，元数据服务器 123 通过发送应用于客户机计算装置 110 的尚未高速缓存到客户机上的所有驱动程序更新来响应。如果满足其先决条件和条件，则在响应 1065 中将驱动程序更新发送到客户机计算装置 110。传递驱动程序更新的响应 1065 较佳地传递每一更新的指令部分。驱动程序更新然

后被写入客户机计算装置的更新高速缓存中。

在接收包含驱动程序更新的响应 1065 之后，同步子例程 1050 发送对所接收的软件和硬件更新的每一个的本地化数据的请求 1066。如上所述，每一更新的本地化数据部分包括描述更新的通用信息。例如，本地化数据部分可包括描述更新的特征和益处的信息。本地化数据部分也可包括更新的安装过程的文本描述。另外，本地化数据部分可包括关于更新的任何其它数据或信息。

由此，在接收了对接收到的软件更新和硬件更新的每一个的本地化数据的请求 1066 之后，元数据服务器 123 通过发送客户机更新高速缓存中保存的所有接收到的软件更新和硬件更新的所有本地化数据来响应。一旦接收到，本地化数据可由软件应用程序来处理，以确定哪些更新需要被安装。或者，接收到的本地化数据可显示给用户，以向用户通知对客户机计算装置 110 可用的所有更新。在一个实施例中，接收到的本地化数据可在网页上显示。在本示例中，第六和第八更新 921 和 931 的本地化数据可以由客户机接收。如果本地化数据被储存在基础更新 911、912、913 和 915 中，则那些更新的本地化数据也由客户机接收。

图 11 示出了网页 1100 的一个示例，它显示了与对客户机可用的更新相关联的本地化数据的示例。为说明的目的，网页 1100 包括更新的第一详细描述 1105 和另一更新的第二详细描述 1106。如图所示，每一更新分别与用于接收更新的用户选择的选择机制 1103 和 1104 相关联。同样如图所示，网页 1100 配备允许用户控制更新的选择到诸如元数据服务器 123 或下载服务器 124 等服务器的传递的控制按钮 1101。

在本发明的一个方面中，客户机执行多个处理以增强网页 1100 的显示。例如，客户机计算装置 110 检查每一更新的本地化数据以确定特定更新是否是高优先级。这一特征可通过在本地化数据或特定更新的其它组件中查找指示特定更新是高优先级或紧急更新的文本来促进。如果客户机计算装置 110 检测到高优先级更新或紧急更新，则客户机在网页 1100 的可视部分中，如页面的顶部显示高优先级更新。另外，客户机可生成可视指示符，如专用的文本消息 1120，指示该更新是高优先级更新。

客户机计算装置 110 也可检查每一更新的本地化数据，以确定特定更新是否需要排他安装，即，更新具有不能与另一更新的安装文件同时安装的安装文件。这一特征可通过在本地化数据或特定更新的其它组件中查找指示特定更新需要排他安装的文本来促进。如果客户机计算装置 110 检测到这一更新，则客户机显示一可

视指示符，例如图 11 所示的文本消息 1122，它具有需要排他安装的更新的描述。

返回到图 7，软件更新例程 700 在框 708 继续，其中客户机计算装置 110 接收更新的选择。如上所述，响应于控制按钮 1101 的激励，对一个或多个更新的选择可由元数据服务器 123 或下载服务器 124 获得。一旦接收到了一个或多个更新的选择，软件更新例程 700 在框 708 继续，处理所选择的软件更新。

依照本发明的又一方面，软件更新服务 120 可提供一种在软件更新服务和客户机计算装置 110 之间选择和发送信息的方法。图 12A 和 12B 所示是依照本发明的软件更新处理子例程 1200，它由客户机计算装置 110 实现，用于检索和安装所请求的软件。如上所述，一旦生成或接收到了对软件更新的选择，就可以实现软件更新处理子例程 1200。参考图 12A，在框 1202,更新管理组件 112 例示更新代理 118。在本发明的一个说明性实施例中，更新代理 118 是一种专用软件组件，用于确定需要什么软件更新信息来完成请求的软件更新、生成更新代理的安装组件的所需要的版本、通过将现有文件与增量补丁合并生成更新文件、和/或启动更新文件的安装。在更新代理 118 已被例示的情况下，可省略框 1202。

在框 1204，更新代理 118 从更新服务 120 获取软件更新信息。在本发明的一个说明性实施例中，更新服务 120 发送的软件更新信息是包的形式，如自提取文件，它包括可由更新代理使用的各种数据。在一个方面，包可包括对应于特定软件更新的所有文件的列表。另外，包可包括补丁存储清单的至少一部分的副本，该清单将要更新的文件的特定版本映射到储存在更新服务 120 上的补丁存储文件中的对应软件更新增量补丁。包也可包括用于要更新的每一文件的安装信息，它可包括完成安装所需要的安装组件的版本的标识。此外，包也可包括用于更新代理 118 的安装组件，或更新已储存在客户机计算装置 110 上的安装组件的版本的增量补丁。再者，包可包括允许更新代理确定软件更新是否成功的验证信息。例如，验证信息可包括更新文件的参考散列值用于比较。更新代理 118 也可验证包的内容。

在判别块 1206，执行测试以确定更新代理 118 是否需要更新安装组件的版本以实现更新。相关领域的技术人员将认识到，在自提取文件中发送安装组件的完整副本可增加更新服务 120 对每一软件更新发送的数据量。因此，在本发明的一个说明性实施例中，可将安装组件的基线版本储存在客户机计算装置中，并通过安装组件增量补丁对当前软件更新的需求来特别更新。因此，自提取文件中的安装信息指令更新代理 118 任何所包括的安全组件更新是否需要与客户机计算装置 110 上的安装组件的基线版本合并。如果需要更新，则在框 1208，更新代理 118 更新基线安

装组件，如后文参考图 13 更详细讨论的。

一旦更新代理更新了安装组件，或者安装组件不需要更新，则在框 1210，更新代理 118 执行安装在计算装置 110 上的文件和文件的特定版本的清查。在本发明的一个说明性实施例中，更新代理 118 可向客户机计算装置 110 文件系统查询在包内被标识为对应于所选择的更新的所有文件。或者，如果更新代理 118 最近已执行了清查，则可使用详细目录的高速缓存的版本。在框 1212，更新代理 118 标识完成所请求的更新所需要什么软件更新信息。在本发明的一个说明性实施例中，包存储清单包括已安装文件的版本到所需要的增量补丁的映射。因此，如果增量打补丁可用，则更新代理 118 将使用该映射来标识特定的增量补丁及其在包存储文件中的偏移位置。或者，如果增量补丁不可用或不能实现，则更新代理 118 可标识整个文件用于下载。

现在参考图 12B，在框 1214，更新数据代理发送对已标识软件更新信息的请求。在本发明的一个说明性实施例中，更新代理 118 可通过指示特定范围的补丁，将对特定增量补丁的请求从补丁存储文件发送到更新服务 120 的下载服务器 124。如上所述，补丁存储文件包括大量的可适用增量补丁，其中每一增量补丁由其在补丁存储文件中的位置来标识。由于补丁存储文件在某些实现中可能相当大，因此更新代理 118 可使用仅请求来自补丁存储文件中的特定位置的数据的请求，如补丁存储清单中所指示的。在本发明的一个替换实施例中，更新代理 118 可请求更新文件的整个副本和/或补丁存储文件的完整副本。

在本发明的一个替换实施例中，可能不排他地与更新服务 120 相关联的另一下载服务器可处理更新代理 118 请求。在这一实施例中，请求补丁存储文件可完整或部分地发送到网络上的任意数量的附加下载服务器。附加下载服务器可以是用于更新专用网络上的客户机的专用网络的一部分。此外，附加下载服务器可以是公共网络的一部分。在专用网络环境中，下载服务器可获取补丁存储文件的完整副本，用于处理客户机请求。或者，下载服务器也可在处理来自其它客户机的先前的数据请求时高速缓存补丁存储文件的各部分，并使用高速缓存数据来满足下载。因此，附加下载服务器可减少对更新服务 120 的下载服务器 124 的通信张力。

在框 1216，更新代理 118 接收所请求的更新信息。在本发明的一个说明性实施例中，所请求的更新信息可以用两种方法发送。在被称为手动更新的第一种方法中，更新请求连同对直接 HTTP 数据传送响应的请求一起发送到更新服务 120。在该方法中，更新服务 120 可使用可用于向更新代理 118 发送所请求数据的整个带宽

的全部。在被称为自动更新的第二种方法中，更新请求连同对间接 HTTP 数据传送响应的请求一起发送到更新服务 120。在这一响应中，更新服务 120 发送所请求的数据作为后台进程。后台进程可以用使用最小量可用带宽的方式来实现。此外，后台进程可在下载过程期间中断，并可在下一可用时刻重新开始。通过后台进程发送所请求数据的系统和方法的描述在 2000 年 2 月 16 日提交的，名为“System and Method for Transferring Data Over a Network（通过网络传输数据的系统和方法）”的共同转让并共同提交的待决美国专利申请号 09/505,735 中有描述，该申请通过引用结合于此。相关领域的技术人员可以理解，前台和后台数据传输不必要反映所选择的软件更新的优先级，但是相反反映了如何分配带宽来获取更新信息。

一旦从更新服务接收到了所请求的信息，在框 1218，更新代理 118 将增量补丁与对应的已安装文件合并。在本发明的一个说明性实施例中，更新代理 118 可高速缓存已安装文件的原始版本，以确保所选择的文件不会在下载和合并过程中改变。此外，高速缓存的已安装文件的原始版本可用于卸载所选择的更新。

在判别框 1220，执行测试以确定更新文件是否有效。在本发明的一个说明性实施例中，更新代理 118 可使用散列算法，以将从更新信息包中获得并对应于有效文件更新的参考散列值与来自当前修改的文件的散列相比较。如果散列不匹配，则当前修改的文件不是有效的。相关领域的技术人员可以理解，可使用多种替换确认算法的任一种。如果更新文件不是有效的，则子例程 1200 返回到框 1214，其中更新代理可再次请求更新信息。或者，如果更新代理 118 已若干次不成功地尝试了生成更新文件，则更新的代理可实现若干后退过程的任一种。在本发明的一个实施例中，更新代理 118 可向更新服务 120 请求储存在补丁存储文件中，并从补丁存储清单标识的更新文件的已完成副本。在本发明的另一实施例中，更新代理 118 可向更新服务 120 请求自主式文件中的更新文件的副本。在本发明的又一实施例中，另一子例程 1200 可以失败。

一旦所选择的文件有效，则在判别块 1222，执行测试以确定是否需要任何额外的下载。在本发明的一个说明性实施例中，子例程 1200 进入一迭代循环，它在完成了先前选择的下载之后连续地核查额外的下载。如果文件的状态在下载期间改变，则更新代理 118 可继续请求对新状态改变的额外下载。如果需要额外的下载，则在框 1224，更新代理 118 执行另一清查，并标识所有可适用的增量补丁。子例程 1200 然后返回到框 1214。

一旦完成了所有的所请求的更新下载，在判别框 1226，执行测试以确定客户

机机器的状态是否改变。在本发明的一个说明性实施例中，在更新文件的更新信息和实际安装的下载和合并之间会经过某些时间。因此，在安装更新文件之前，更新代理确定客户机计算装置状态是否改变。如果状态已改变，则文件更新可能不是有效的，并且更新在框 1228 失败。或者，如果没有发生状态改变，则更新代理 118 在框 1230 安装更新文件，并且子例程 1200 返回到框 1232。

现在参考图 13，描述了由客户机计算装置 110 实现，用于更新对应于框 1208（图 12A）的基线安装组件的子例程 1300。在判别框 1302，执行测试以确定从更新服务 120 发送到更新代理 118 的自提取文件中是否包括了新的基线安装组件。在本发明的一个说明性实施例中，如果更新基线安装程序所需的增量补丁在大小上可与已更新安装组件的发送比较，则将发送新的基线安装组件。如果包括了已更新安装组件，则在框 1304，更新代理安装已更新的基线安装组件作为新的安装组件。另外，新的已更新安装组件可以保存在客户机计算装置 110 的存储器中，以担当用于附加更新的基线安装程序。在框 1306，子例程返回。

如果在自提取文件中未包括已更新的基线安装组件，则在框 1308，更新代理 118 从自提取文件获取基线安装组件增量补丁。在本发明的一个说明性实施例中，基线安装组件增量补丁对应于可与基线安装组件合并来生成已更新基线安装组件的软件代码。因此，在框 1310，更新代理将基线安装组件增量补丁与基线安装组件合并。在框 1312，更新代理 118 然后将已更新基线安装组件制定为当前安装组件。在本发明的一个说明性实施例中，已更新安装组件在安装完成之后将不被保存。依照该实施例，更新代理 118 仅维护客户机计算装置 110 的存储器中有限数量的基线安装组件。因此，更新代理在每一次安装时生成临时的已更新安装组件。由于每一客户机计算装置 110 只能对应于有限数量的基线安装组件，因此仅需要更新服务 120 对每一客户机计算装置发送单个基线安装组件增量补丁。在框 1314，子例程 1300 返回。

尽管示出并描述了本发明的较佳实施例，然而可以理解，可在不脱离本发明的精神和范围的情况下作出对其作出各种改变。例如，尽管此处描述的说明性示例应用于软件更新，然而本发明的范围包括除关于软件更新的信息的分发和传递之外的其它使用。因此，除非在本揭示中明确地排除了特定主题，可以理解，本发明的范围应用于不同于软件更新或除此之外的任何类型数据的分发和传递。

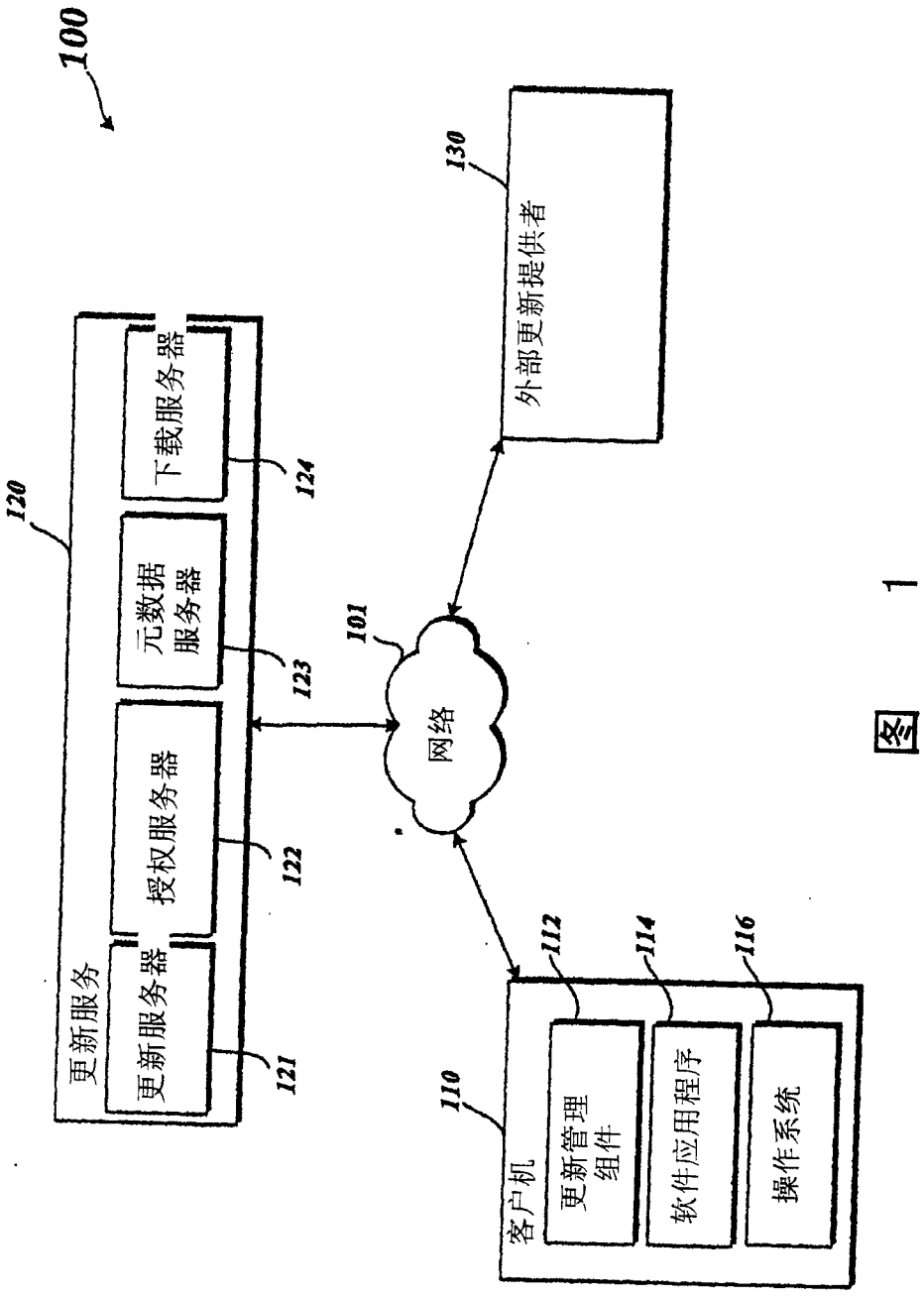

100
更新服务
更新服务器
授权服务器
元数据服务器
下载服务器
120
121
122
123
124
网络
101
外部更新提供者
130
客户机
更新管理组件
软件应用程序
操作系统
110
112
114
116

图 1

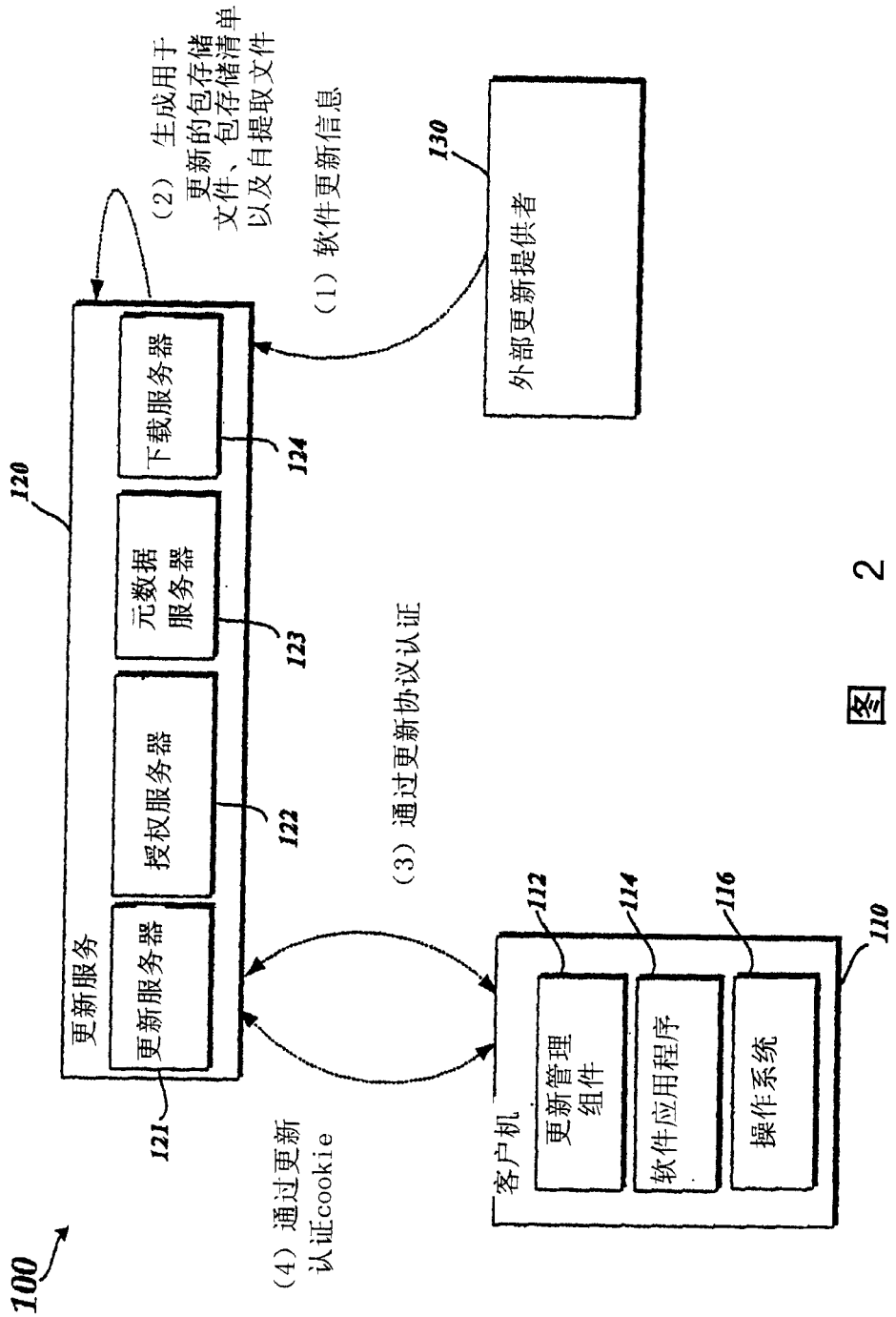

100
更新服务
更新服务器
授权服务器
元数据服务器
下载服务器
120
121
122
123
124
(1) 软件更新信息
(2) 生成用于更新的包存储文件、包存储清单以及自提取文件
(3) 通过更新协议认证
(4) 通过更新认证cookie
外部更新提供者
130
客户机
更新管理组件
软件应用程序
操作系统
112
114
116
110

图 2

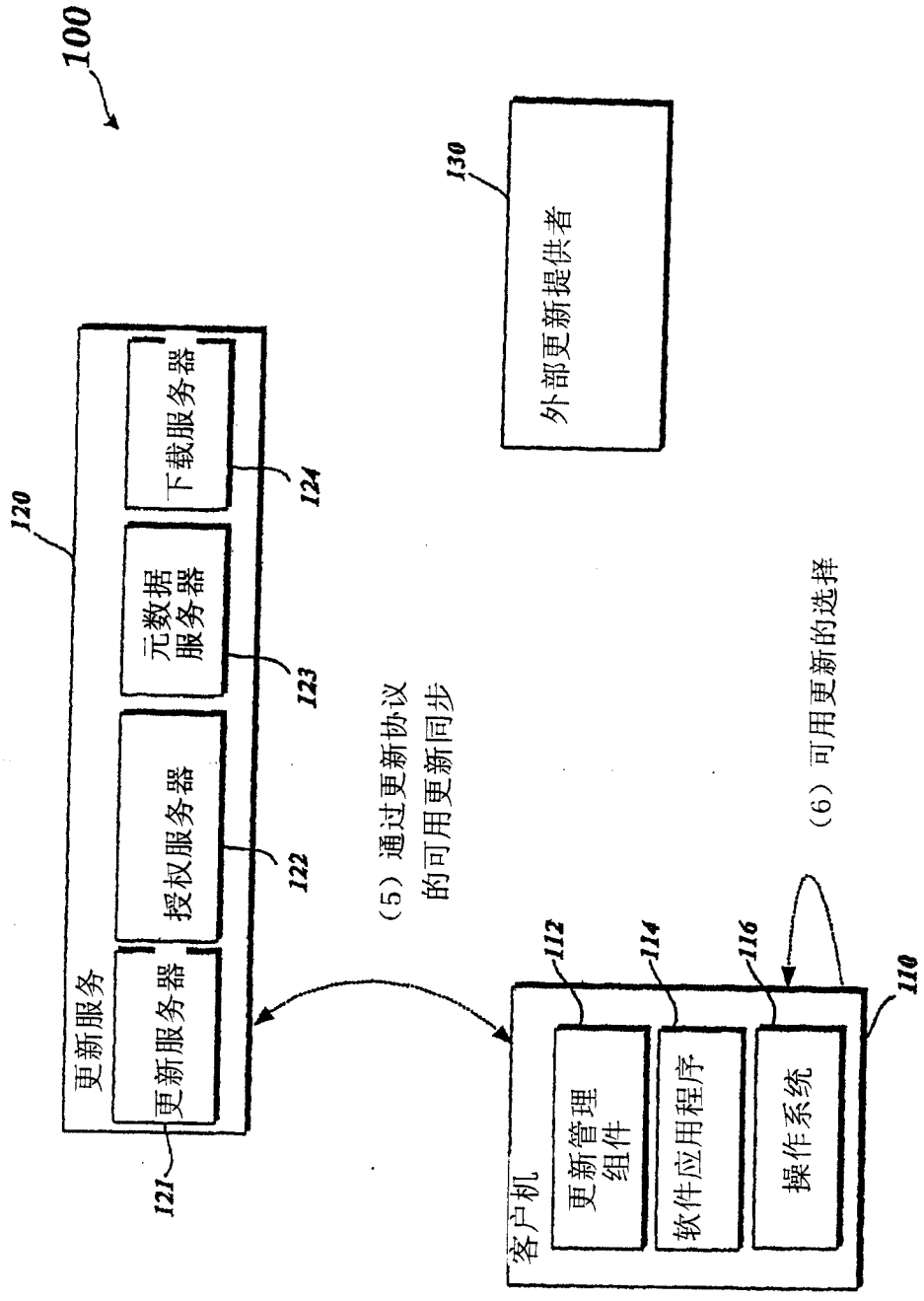

100
外部更新提供者
130
更新服务
更新服务器
授权服务器
元数据服务器
下载服务器
120
121
122
123
124
(5) 通过更新协议的可用更新同步
(6) 可用更新的选择
客户机
更新管理组件
软件应用程序
操作系统
112
114
116
110

图 3

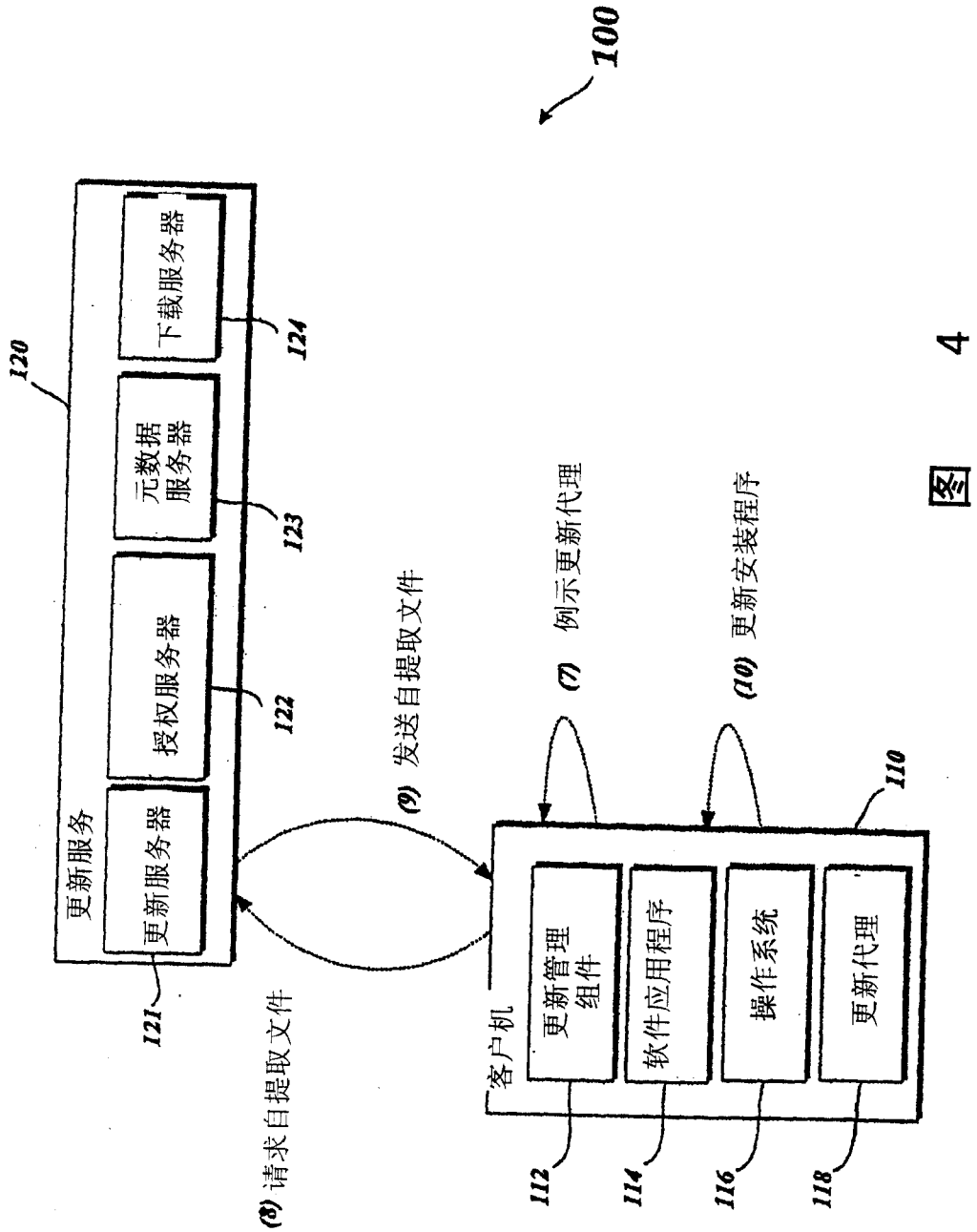

100
120
更新服务
更新服务器
授权服务器
元数据服务器
下载服务器
121
122
123
124
(8) 请求自提取文件
(9) 发送自提取文件
(7) 例示更新代理
(10) 更新安装程序
客户机
110
更新管理组件
软件应用程序
操作系统
更新代理
112
114
116
118

图 4

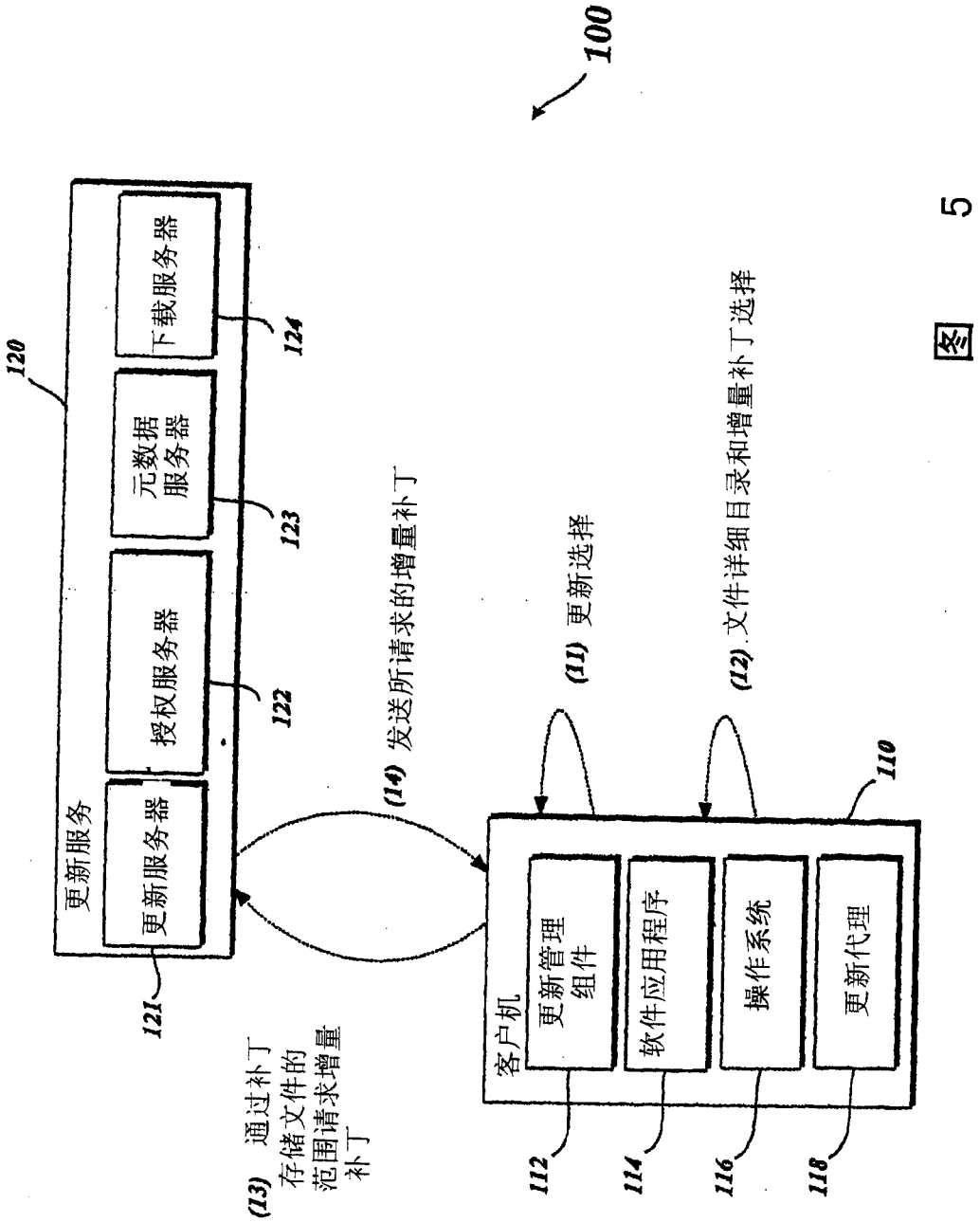

100
更新服务
更新服务器
授权服务器
元数据服务器
下载服务器
120
121
122
123
124
(13) 通过补丁存储文件的范围请求增量补丁
(14) 发送所请求的增量补丁
客户机
更新管理组件
软件应用程序
操作系统
更新代理
110
112
114
116
118
(11) 更新选择
(12) 文件详细目录和增量补丁选择

图 5

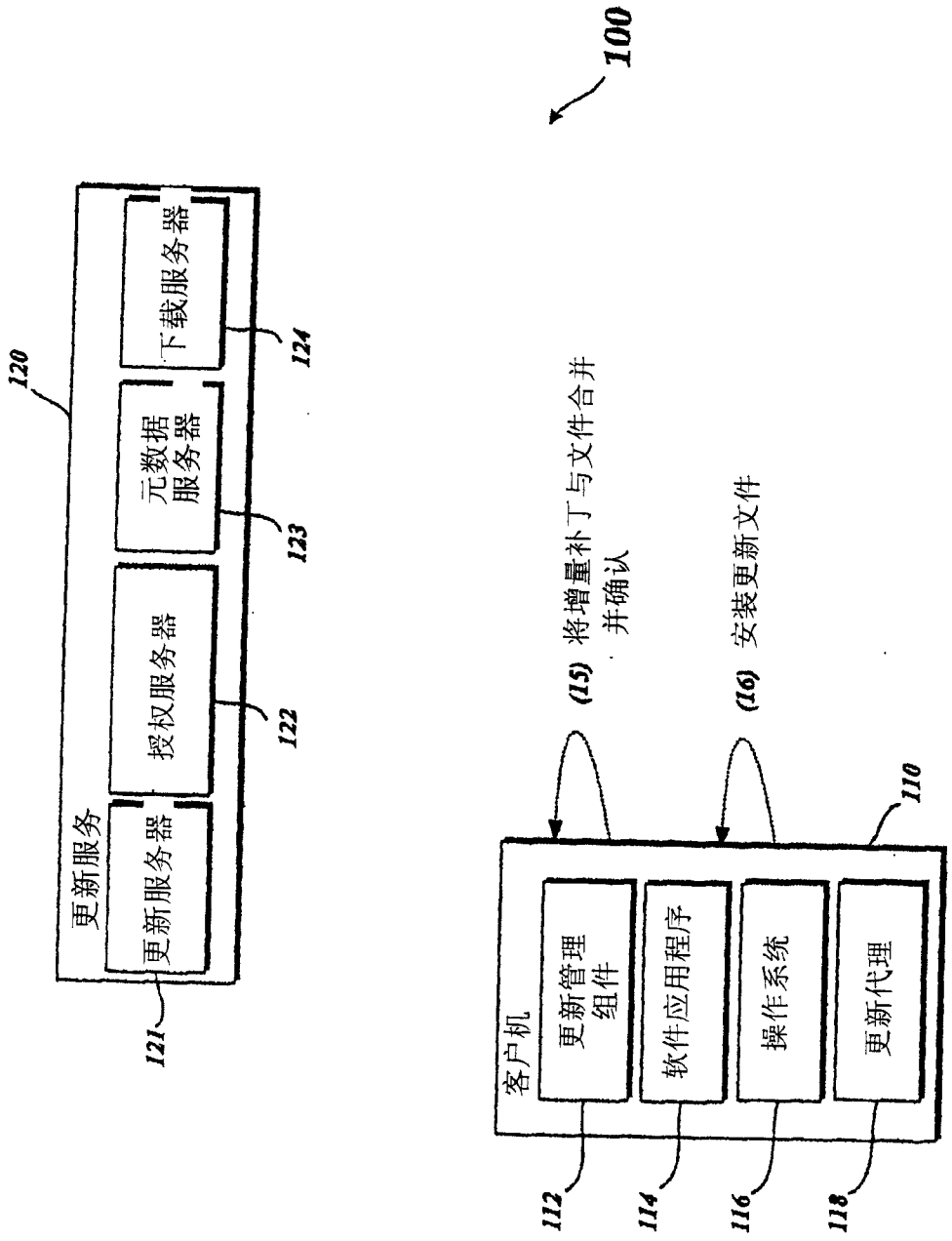

100
更新服务
更新服务器
授权服务器
元数据服务器
下载服务器
120
121
122
123
124
客户机
更新管理组件
软件应用程序
操作系统
更新代理
110
112
114
116
118
(15) 将增量补丁与文件合并并确认
(16) 安装更新文件

图 6

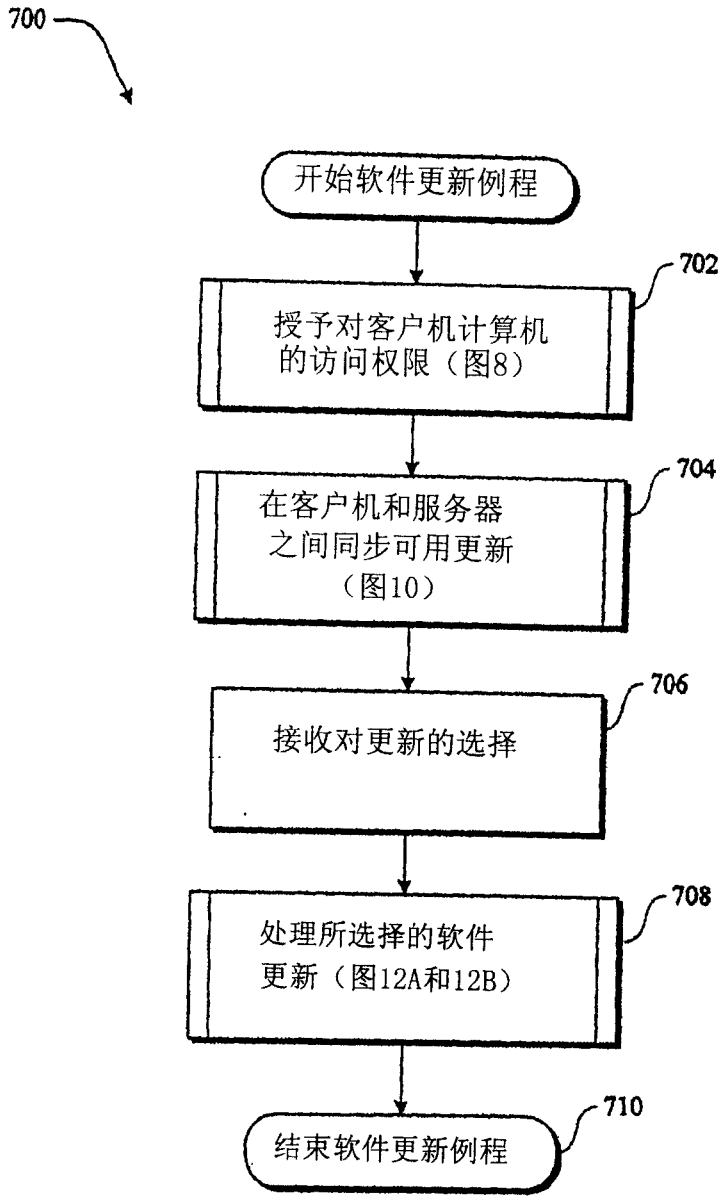

700
开始软件更新例程
授予对客户机计算机的访问权限（图8）
702
在客户机和服务器之间同步可用更新（图10）
704
接收对更新的选择
706
处理所选择的软件更新（图12A和12B）
708
结束软件更新例程
710

图 7

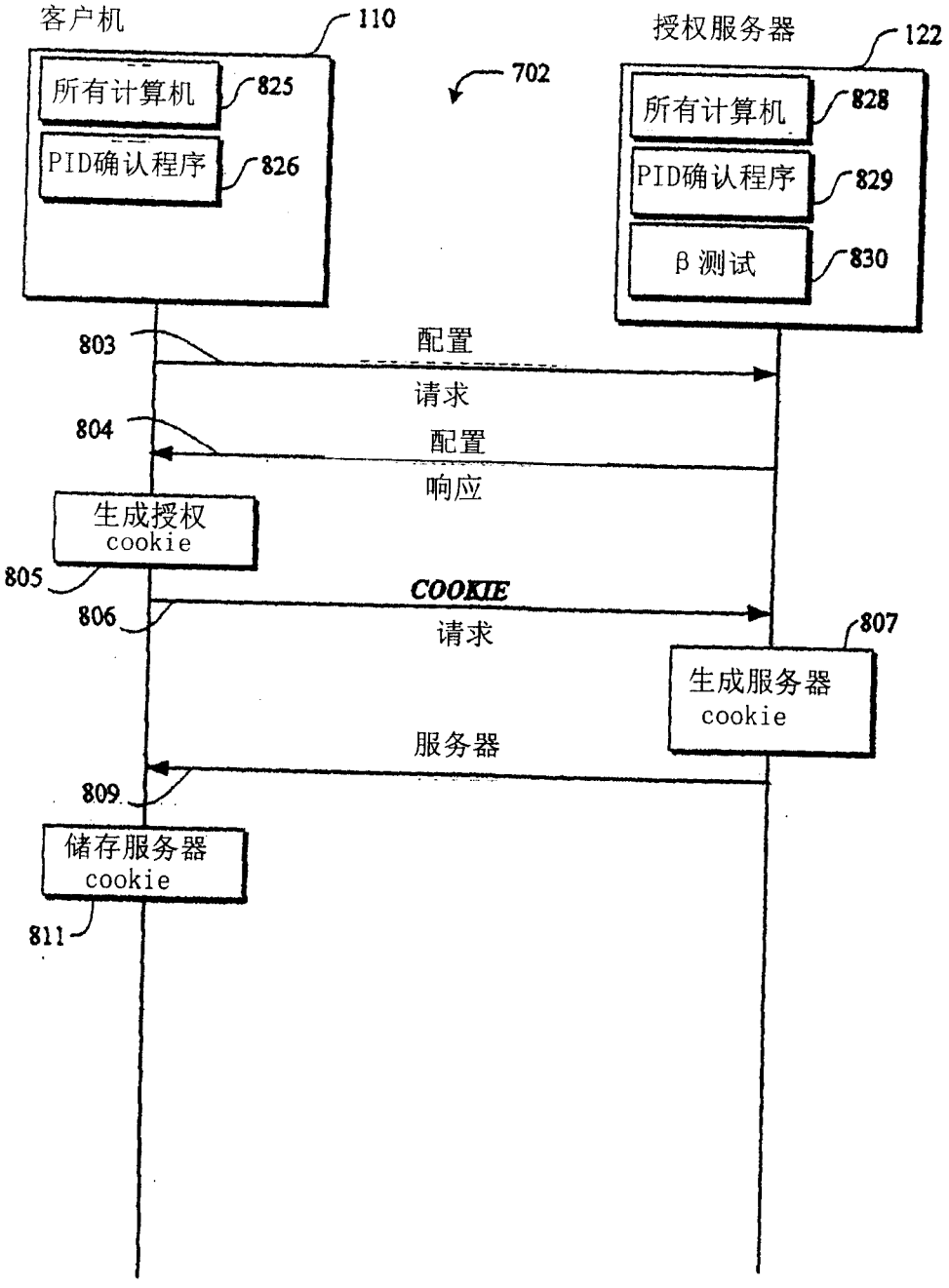

客户机
110
所有计算机
825
PID确认程序
826
702
授权服务器
122
所有计算机
828
PID确认程序
829
β测试
830
803
配置
请求
804
配置
响应
生成授权
cookie
805
COOKIE
806
请求
807
生成服务器
cookie
服务器
809
储存服务器
cookie
811

图 8

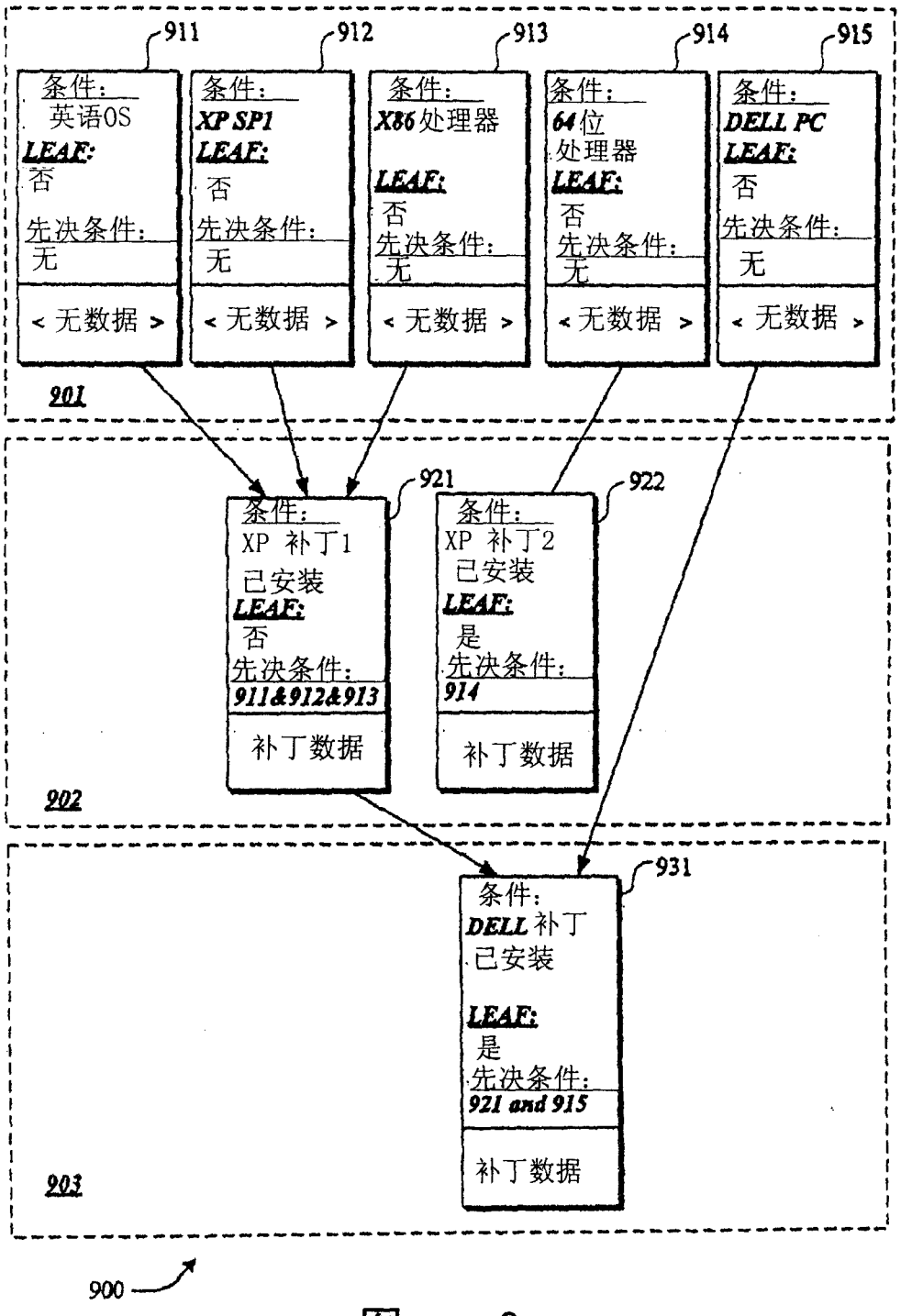

911
912
913
914
915
条件：
英语OS
LEAF:
否
先决条件：
无
< 无数据 >
条件：
XP SP1
LEAF:
否
先决条件：
无
< 无数据 >
条件：
X86处理器
LEAF:
否
先决条件：
无
< 无数据 >
条件：
64位
处理器
LEAF:
否
先决条件：
无
< 无数据 >
条件：
DELL PC
LEAF:
否
先决条件：
无
< 无数据 >
901
921
922
条件：
XP 补丁1
已安装
LEAF:
否
先决条件：
911&912&913
补丁数据
条件：
XP 补丁2
已安装
LEAF:
是
先决条件：
914
补丁数据
902
931
条件：
DELL补丁
已安装
LEAF:
是
先决条件：
921 and 915
补丁数据
903
900

图 9

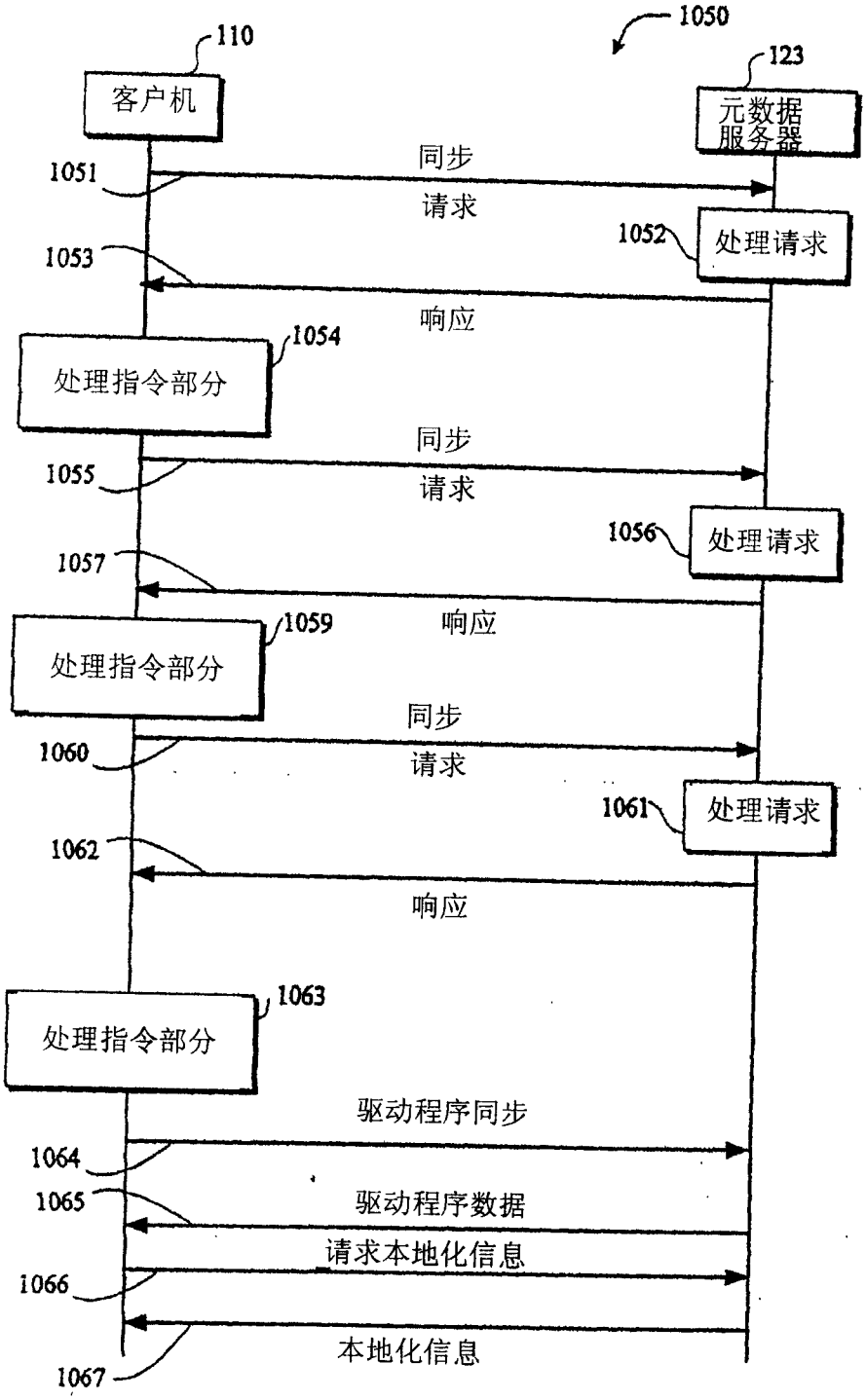

1050
110
客户机
123
元数据服务器
同步
1051
请求
1052
处理请求
1053
响应
1054
处理指令部分
同步
1055
请求
1056
处理请求
1057
响应
1059
处理指令部分
同步
1060
请求
1061
处理请求
1062
响应
1063
处理指令部分
驱动程序同步
1064
1065
驱动程序数据
请求本地化信息
1066
本地化信息
1067

图 10

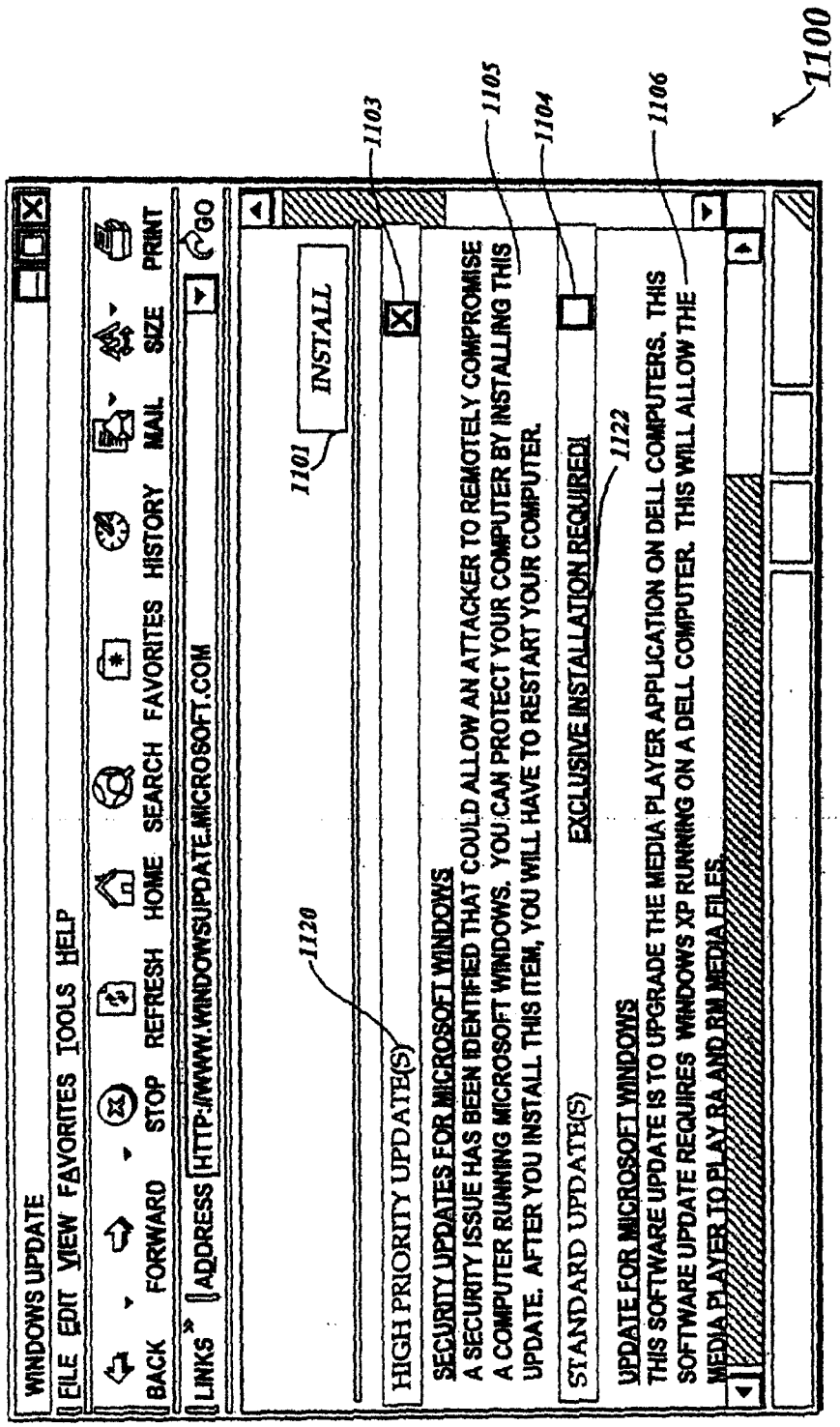

WINDOWS UPDATE
FILE EDIT VIEW FAVORITES TOOLS HELP
BACK FORWARD STOP REFRESH HOME SEARCH FAVORITES HISTORY MAIL SIZE PRINT
LINKS ADDRESS HTTP://WWW.WINDOWSUPDATE.MICROSOFT.COM GO
INSTALL
1101
1103
1105
1104
1106
1100
1120
HIGH PRIORITY UPDATE(S)
SECURITY UPDATES FOR MICROSOFT WINDOWS
A SECURITY ISSUE HAS BEEN IDENTIFIED THAT COULD ALLOW AN ATTACKER TO REMOTELY COMPROMISE A COMPUTER RUNNING MICROSOFT WINDOWS. YOU CAN PROTECT YOUR COMPUTER BY INSTALLING THIS UPDATE. AFTER YOU INSTALL THIS ITEM, YOU WILL HAVE TO RESTART YOUR COMPUTER.
STANDARD UPDATE(S)
EXCLUSIVE INSTALLATION REQUIRED!
1122
UPDATE FOR MICROSOFT WINDOWS
THIS SOFTWARE UPDATE IS TO UPGRADE THE MEDIA PLAYER APPLICATION ON DELL COMPUTERS. THIS SOFTWARE UPDATE REQUIRES WINDOWS XP RUNNING ON A DELL COMPUTER. THIS WILL ALLOW THE MEDIA PLAYER TO PLAY RA AND RM MEDIA FILES.

图 11

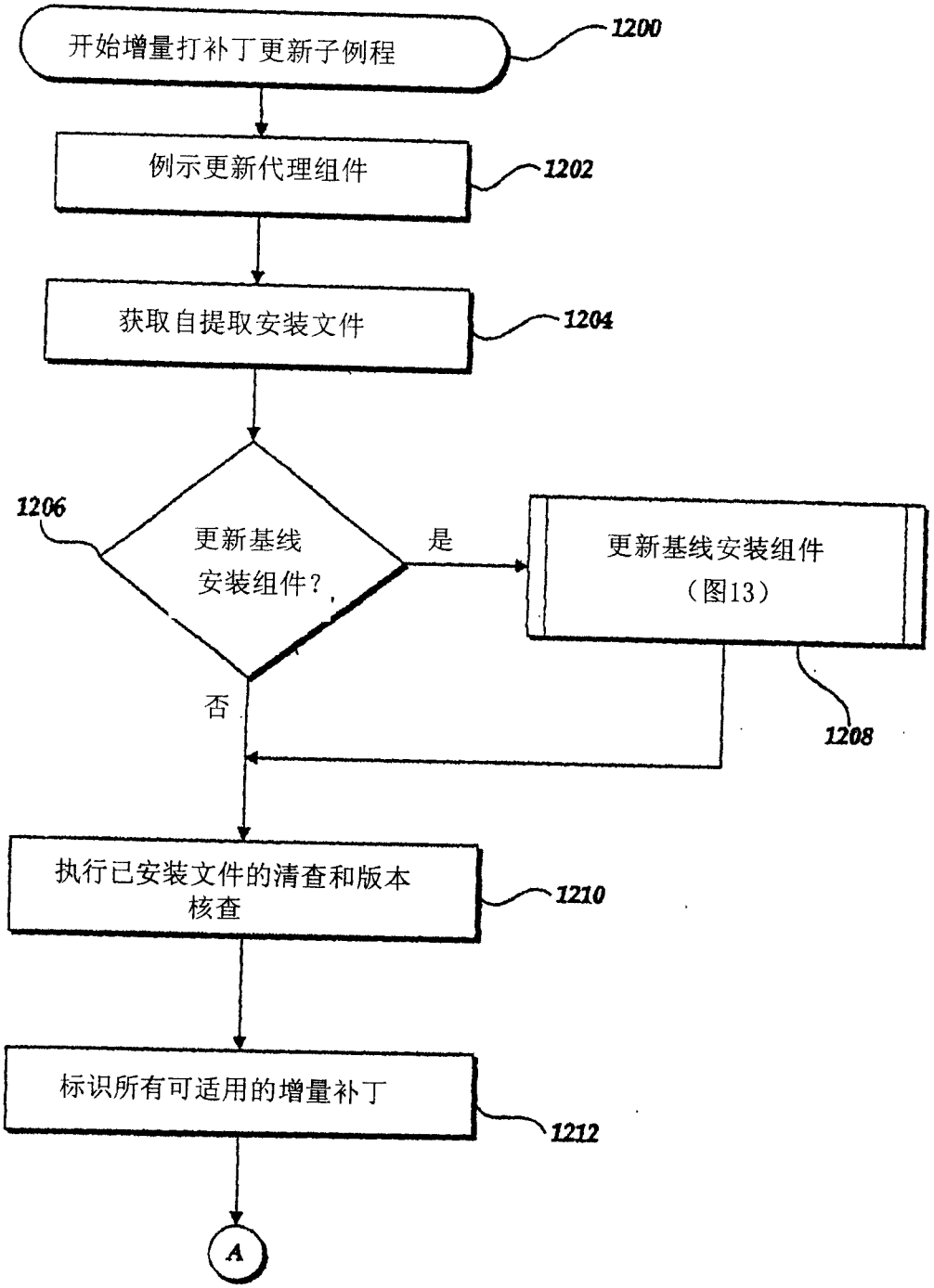

开始增量打补丁更新子例程
1200
例示更新代理组件
1202
获取自提取安装文件
1204
1206
更新基线
安装组件？
是
更新基线安装组件
（图13）
1208
否
执行已安装文件的清查和版本
核查
1210
标识所有可适用的增量补丁
1212
A

图 12A

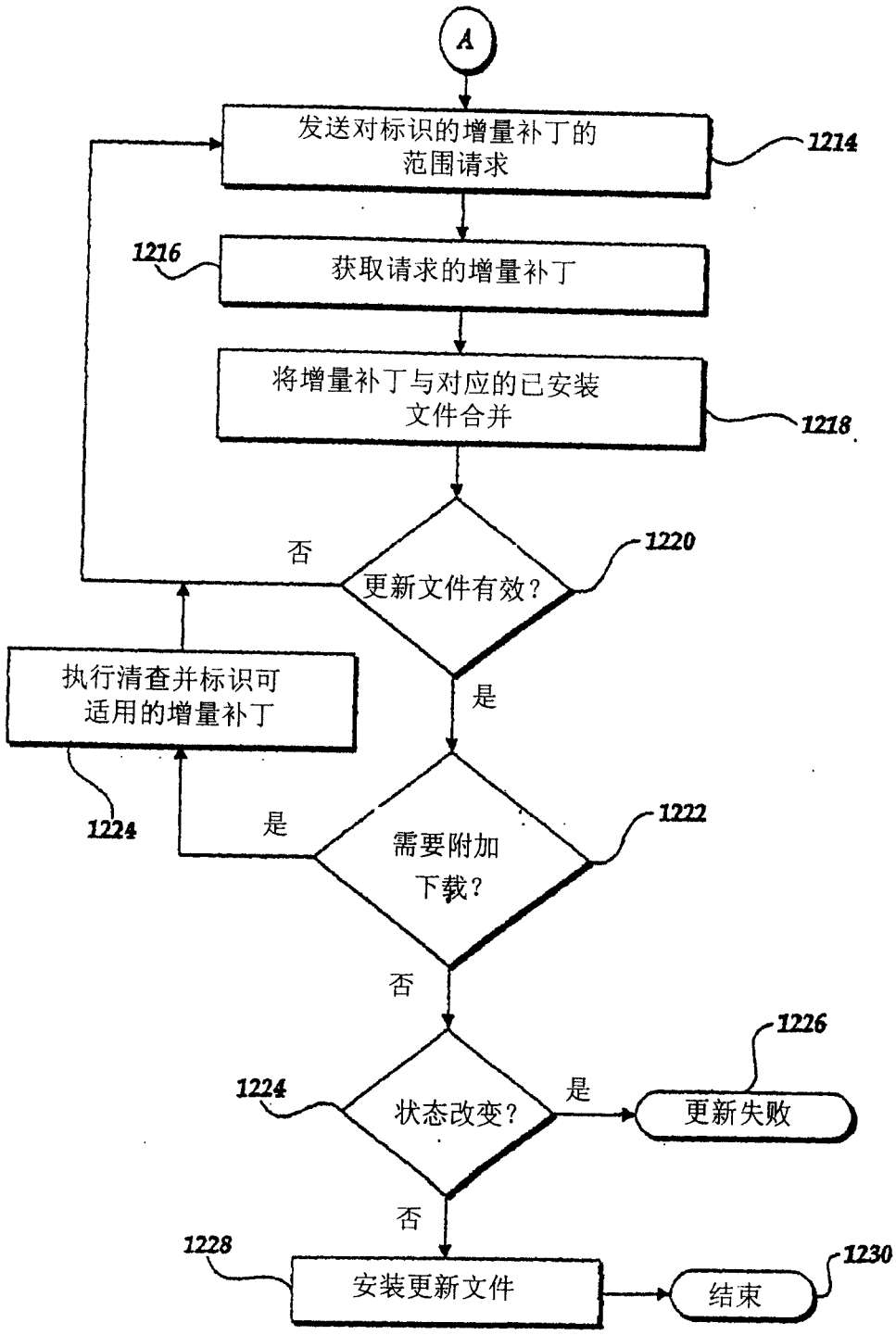

A
发送对标识的增量补丁的范围请求
1214
1216
获取请求的增量补丁
将增量补丁与对应的已安装文件合并
1218
否
更新文件有效?
1220
是
执行清查并标识可适用的增量补丁
1224
是
需要附加下载?
1222
否
1224
状态改变?
是
1226
更新失败
否
1228
安装更新文件
结束
1230

图 12B

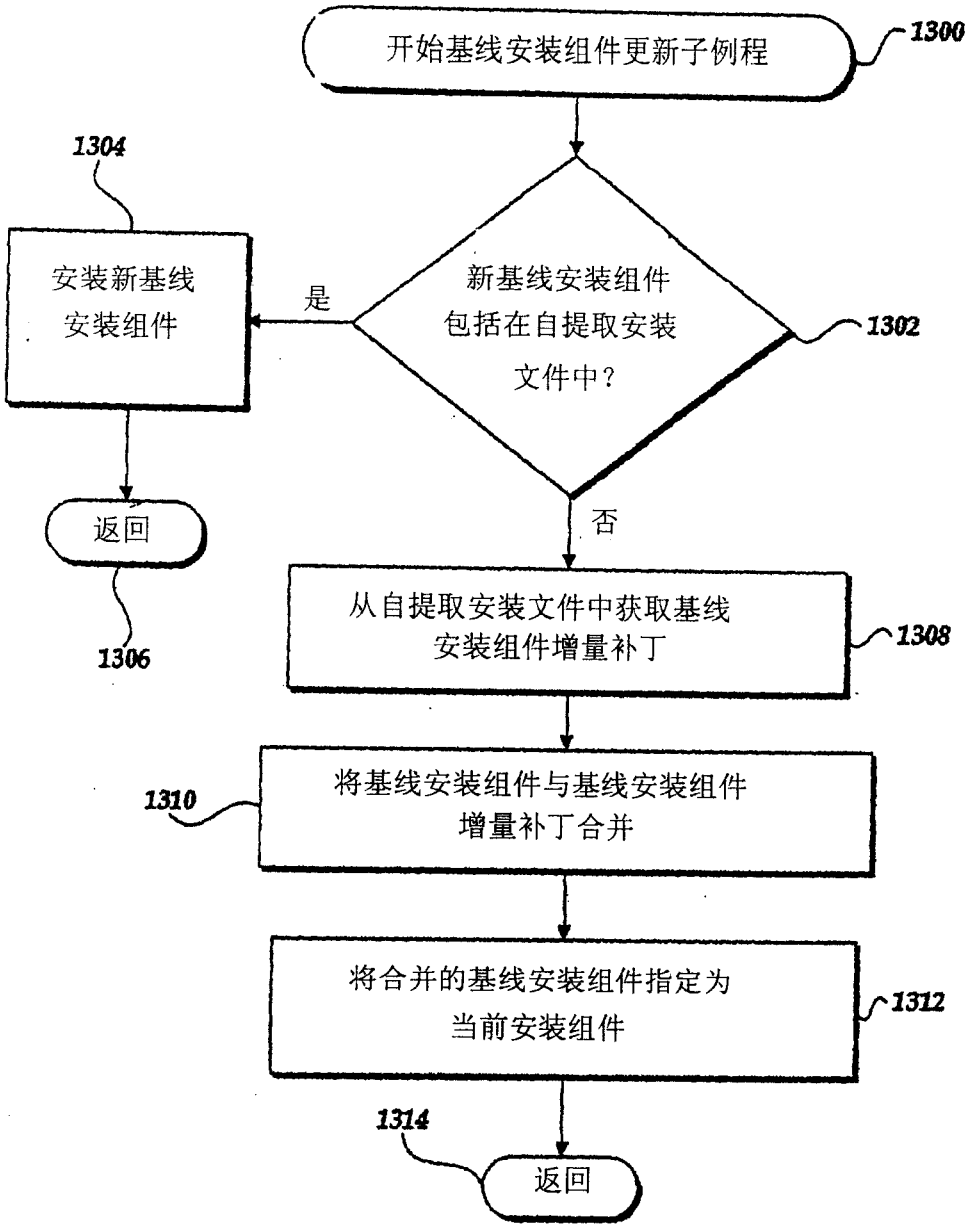

开始基线安装组件更新子例程
1300
新基线安装组件包括在自提取安装文件中？
1302
是
否
1304
安装新基线安装组件
返回
1306
从自提取安装文件中获取基线安装组件增量补丁
1308
1310
将基线安装组件与基线安装组件增量补丁合并
将合并的基线安装组件指定为当前安装组件
1312
1314
返回

图 13