

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.

G06F 15/00 (2006.01)

G06F 15/16 (2006.01)

G06F 9/44 (2006.01)



[12] 发明专利申请公布说明书

[21] 申请号 200780015979.8

[43] 公开日 2009年5月20日

[11] 公开号 CN 101438261A

[22] 申请日 2007.5.4

[21] 申请号 200780015979.8

[30] 优先权

[32] 2006.5.5 [33] US [31] 11/418,747

[86] 国际申请 PCT/US2007/010873 2007.5.4

[87] 国际公布 WO2007/130594 英 2007.11.15

[85] 进入国家阶段日期 2008.11.3

[71] 申请人 微软公司

地址 美国华盛顿州

[72] 发明人 J·K·班克斯顿 Z·王

C·M·鲁塞尔 J·M·菲尔莫尔

K·杰拉斯 K·钦特里斯

[74] 专利代理机构 上海专利商标事务所有限公司

代理人 陈斌

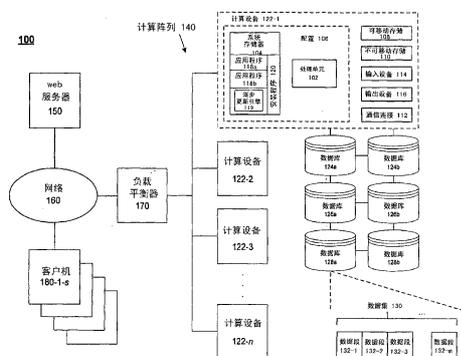
权利要求书 3 页 说明书 15 页 附图 5 页

[54] 发明名称

执行逐步升级的技术

[57] 摘要

描述了执行逐步升级的技术。装置可以包括具有应用程序的第一版本的数据集的数据库，和耦合到该数据库的计算设备。该计算设备可以具有处理器和存储器，存储器存储应用程序的第一版本、安装程序、和逐步更新引擎。处理器可以执行安装程序来安装应用程序的第二版本，且所述逐步更新引擎一次一个数据段地将数据集转换到第二版本所使用的数据库模式。对其它实施方式也予以描述并主张权利。



1. 一种方法，包括：
安装应用程序的新版本（202）；以及
一次一个数据段地将所述应用程序的先前版本的数据集转换到所述新版本所使用的数据模式。
2. 如权利要求 1 所述的方法，其特征在于，包括：
将所述先前版本的第一数据段转换到所述数据模式；以及
在转换所述第一数据段的同时提供对所述先前版本的第二数据段的访问。
3. 如权利要求 1 所述的方法，其特征在于，包括：
将所述先前版本的第一数据段转换到所述数据模式；以及
在所述转换期间阻止对所述第一数据段的访问。
4. 如权利要求 1 所述的方法，其特征在于，包括：
标识所述先前版本的域名；
创建所述先前版本的临时域名；
将所述域名重命名为所述临时域名；以及
将使用所述域名的访问请求重定向到所述临时域名。
5. 如权利要求 1 所述的方法，其特征在于，包括：
标识所述先前版本的域名；以及
一旦已经转换了所述数据集，则使用所述新版本的所述域名。
6. 如权利要求 1 所述的方法，其特征在于，包括与其它应用程序接口以与转换所述先前版本的所述数据集相一致地转换对应的数据集。
7. 如权利要求 1 所述的方法，其特征在于，包括选择具有来自一个或多个内容数据库的、web 应用程序所使用的数据的数据段。
8. 如权利要求 1 所述的方法，其特征在于，包括选择具有来自一个或多个内容数据库的、一组分层 web 应用程序所使用的数据的数据段。
9. 一种物品，包括含有指令的机器可读存储介质，所述指令如果被执行则使系统安装应用程序的新版本（202），并一次一个数据段地将所述应用程序的先前版本的数据集转换到所述新版本所使用的数据模式（204）。

10. 如权利要求 9 所述的物品，其特征在于，还包括如果被执行则使所述系统将所述先前版本的第一数据段转换到所述数据模式、并在转换所述第一数据段的同时提供对所述先前版本的第二数据段的访问的指令。

11. 如权利要求 9 所述的物品，其特征在于，还包括如果被执行则使所述系统将所述先前版本的第一数据段转换到所述数据模式、并在所述转换期间阻止对所述第一数据段的访问的指令。

12. 如权利要求 9 所述的物品，其特征在于，还包括如果被执行则使所述系统标识所述先前版本的域名、创建所述先前版本的临时域名、将所述域名重命名为所述临时域名、并将使用所述域名的访问请求重定向到所述临时域名的指令。

13. 如权利要求 9 所述的物品，其特征在于，还包括如果被执行则使所述系统标识所述先前版本的域名、并且一旦已经转换了所述数据集则使用所述新版本的所述域名的指令。

14. 如权利要求 9 所述的物品，其特征在于，还包括如果被执行则使所述系统选择具有来自一个或多个内容数据库的、web 应用程序所使用的数据的数据段的指令。

15. 如权利要求 9 所述的物品，其特征在于，所述机器可读存储介质包括计算设备、计算机可读介质、通信介质、或传播的信号。

16. 一种装置，包括：

具有应用程序的第一版本的数据集（130）的数据库（128）；

耦合到所述数据库的计算设备（122），所述计算设备具有处理器（102）和存储器（104），所述存储器存储应用程序的第一版本（118a）、安装程序（120）、和逐步更新引擎（119）；以及

所述处理器执行所述安装程序来安装所述应用程序的第二版本（118b），且所述逐步更新引擎一次一个数据段（132）地将所述数据集转换到所述第二版本所使用的数据模式。

17. 如权利要求 16 所述的装置，其特征在于，所述逐步更新引擎在转换一个数据段的同时提供对另一个数据段的访问。

18. 如权利要求 16 所述的装置，其特征在于，所述逐步更新引擎在转换期间锁定数据段以阻止对所述数据段的写访问。

19. 如权利要求 16 所述的装置，其特征在于，所述逐步更新引擎在转换操作期间将对所述第一版本的第一域名的访问请求重定向到所述第一版本的第二域名。

20. 如权利要求 16 所述的装置，其特征在于，所述逐步更新引擎选择具有来自一个或多个内容数据库的、web 应用程序所使用的数据的数据段。

执行逐步升级的技术

背景

软件应用程序周期性地发布具有改进的特征的新版本。将软件应用程序升级到新版本可能涉及将用户数据从现有数据模式转换到新版本所使用的新的数据模式。在某些情况下，用户数据的量可能过大以至于升级操作可能使设备或网络对于延长的时间段变得不可用。例如，基于服务器的软件应用程序可能具有万亿字节级的用户数据，其可能花几天或甚至几周来升级到新的数据模式。关键任务商业系统或高可用性系统可能发现这种延长的停机时间段是不可接受的。因此，可能需要改进升级技术来解决这些和其它问题。

概述

提供本概述以便以简化的形式介绍将在以下详细描述中进一步描述的一些概念。本概述并不旨在标识出所要求保护的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的主题的范围。

各实施例一般可以针对将软件应用程序逐步升级到该软件应用程序的新版本的升级技术。在一些实施例中，可以使用逐步升级技术来分阶段地更新应用程序的各部分和/或该应用程序的数据集。例如，在一实施例中，安装程序可被安排成在与应用程序的先前版本相同或类似的硬件平台上安装该应用程序的新版本。逐步更新引擎可被安排成将该应用程序的先前版本的数据集转换到该应用程序的新版本所使用的新的数据模式。更具体地，逐步更新引擎可以一次一个数据段地、分阶段地转换数据集。以此方式，在转换操作期间客户机可以访问先前版本的其它数据段，藉此授予对该应用程序的受限的使用而不是致使该应用程序完全不可用。对其它实施方式也予以描述并主张权利。

附图

图 1 示出网络的示例性实施例。

图 2 示出第一逻辑流程的示例性实施例。

图 3 示出第二逻辑流程的示例性实施例。

图 4 示出第三逻辑流程的示例性实施例。

图 5 示出第四逻辑流程的示例性实施例。

详细描述

各实施例一般可以针对软件应用程序的升级技术。具体而言，各实施例可以针对逐步升级技术，其中可以分阶段地升级应用程序以减少停止时间和对客户机和/或用户的潜在影响。这对于例如使用诸如被用来提供 web 主存服务的那些等基于服务器的 web 应用程序的大规模公司网络可能尤其重要。此外，在升级过程期间，逐步升级技术可以允许客户机继续访问基于服务器的 web 应用程序的各部分。这种受限的访问对于关键任务和高可用性应用程序可能特别重要。结果，在升级设备或网络的软件应用程序时，网络管理员可以体验减少了的任务和耗费，且用户可以体验向经升级的应用程序和服务的更无缝的转换。

图 1 示出网络的一示例性实施例。图 1 示出网络 100 的框图。网络 100 可以表示适于实现各种实施例的通用网络体系结构。网络 100 可以包括多个元素。元素可以包括被安排来执行特定操作的任何物理或逻辑结构。视给定的一组设计参数或性能约束的需要，每一元素可被实现为硬件、软件、或其任意组合。硬件元素的示例可以包括处理器、微处理器、电路、电路元件（例如，晶体管、电阻器、电容器、电感器等）、集成电路、专用集成电路（ASIC）、可编程逻辑器件（PLD）、数字信号处理器（DSP）、现场可编程门阵列（FPGA）、存储器单元、逻辑门、寄存器、半导体设备、芯片、微芯片、芯片组等。软件的示例可以包括任何软件组件、程序、应用软件、计算机程序、应用程序、系统程序、机器程序、操作系统程序、中间件、固件、软件模块、例程、子例程、函数、方法、接口、软件接口、应用程序接口（API）、指令集、计算代码、计算机代码、代码段、计算机代码段、文字、值、符号、或其任意组合。虽然图 1 中示出的网络 100 在特定拓扑中具有有限数量的元素，但可以理解，网络 100 在给定实现所需的另选拓扑中可以包括更多或更少元素。各实施例不限于该上下文。

如图 1 所示，例如，网络 100 可以包括各种元素，如计算阵列 140、服务器 150、网络 160、负载均衡器 170、和客户机设备 180-1-s。在一实施例中，例如，服务器 150 可以作为 web 服务器来实现。web 服务器可以包括能够接受

来自客户机设备（例如，客户机 180-1-s 和/或计算设备 122-1-n）的超文本传输协议（HTTP）请求以提供网页的计算设备。网页通常是使用诸如超文本标记语言（HTML）、可扩展标记语言（XML）、可扩展超文本标记语言（XHTML）、MICROSOFT WORD®标记语言（WordML）等某种形式的标记语言生成的文档。网络 160 可以包括使用诸如传输控制协议和网际协议（TCP/IP）等一种或多种因特网协议的分组网络。负载均衡器 170 可以包括以计算资源被以高效的方式使用的方式向一组联网的计算机服务器（例如，计算阵列 140）分派工作负载的设备。负载均衡器 170 可以使用例如计算机、服务器、虚拟服务器、网络设备等等来实现。

在各实施例中，计算阵列 140 可以包括多个计算设备 122-1-n。计算设备 122-1-n 的示例可以包括，但不必限于，计算机、计算机系统、计算机子系统、工作站、终端、服务器、web 服务器、虚拟服务器、个人计算机（PC）、台式计算机、膝上型计算机、超膝上型计算机、便携式计算机、手持式计算机、个人数字助理（PDA）、移动计算设备、和任何其它电子、机电或电气设备。各实施例不限于该上下文。

例如，在一实施例中，计算阵列 140 可被实现为服务器场，其中计算设备 122-1-n 的每一个都表示一个服务器、虚拟服务器、虚拟机、单板计算机（SBC）、服务器刀片、web 服务器等。服务器场一般是通常由一个公司维护以实现超出一个机器的能力的服务器需求的计算机服务器的集合。通常，服务器场将主服务器和后备服务器两者同时分配给单个任务，以便在主服务器发生故障的情况下，后备服务器将接管主服务器的功能。例如，服务器场通常被用来提供 web 主存服务。web 主存服务是向个人和组织提供在线系统以供存储信息、图像、视频、音频、文本、动画、电影、图片或可经由万维网（WWW 或“web”）访问的任何其它形式的 web 内容等的一类因特网主存服务。web 主机是在其拥有的服务器上提供空间以供其客户机使用以及提供因特网连接的公司，通常位于数据中心中。web 主机还可以为其不拥有的、位于其数据中心中的服务器提供数据中心空间和到因特网的连接。

图 1 还示出计算设备 122-1 的更详细的框图。计算设备 122-1 可以表示任何计算设备 122-1-n。在其最基本的配置 106 中，计算设备 122-1 通常包括至少

一个处理单元 102 和存储器 104。存储器 104 可以使用机器可读的或计算机可读的、能够存储数据的任何介质来实现，包括易失性和非易失性存储器。例如，存储器 104 可以包括只读存储器（ROM）、随机存取存储器（RAM）、动态 RAM（DRAM）、双数据率 DRAM（DDRAM）、同步 DRAM（SDRAM）、静态 RAM（SRAM）、可编程 ROM（PROM）、可擦除可编程 ROM（EPROM）、电可擦除可编程 ROM（EEPROM）、闪存、诸如铁电聚合物存储器等聚合物存储器、奥氏存储器、相变或铁电存储器、硅-氧化物-氮化物-氧化物-硅（SONOS）存储器、磁卡或光卡、或适于存储信息的任何其它类型的介质。如图 1 所示，存储器 104 可以存储各种软件程序，如一个或多个应用程序 118、逐步更新引擎（GUE）119、安装程序 120、和附带数据。

计算设备 122-1 还可以具有在配置 106 之外的另外的特征和/或功能。例如，计算设备 122-1 可以包括可移动存储 108 和不可移动存储 110，这些存储还可以包括如前所述的各种类型的机器可读的或计算机可读的介质。计算设备 122-1 还可以具有一个或多个输入设备 114，如键盘、鼠标、笔、语音输入设备、触摸输入设备等。诸如显示器、扬声器、打印机等一个或多个输出设备 116 也可以被包括在计算设备 122-1 中。

计算设备 122-1 还可以包括允许计算设备 122-1 与其它设备进行通信的一个或多个通信连接 112。通信连接 112 可以包括各种类型的标准通信元件，如通信接口、网络接口、网络接口卡（NIC）、无线电、无线发射机/接收机（收发机）、有线和/或无线通信介质、物理连接器等。通信介质通常以诸如载波或其它传输机制等已调制数据信号来体现计算机可读指令、数据结构、程序模块或其它数据，并包括任意信息传送介质。术语“已调制数据信号”指的是以在信号中编码信息的方式设定或更改其一个或多个特征的信号。作为示例而非局限，通信介质包括有线通信介质和无线通信介质。有线通信介质的示例可以包括导线、电缆、金属线、印刷电路板（PCB）、背板、交换光纤、半导体材料、双绞线、同轴电缆、光纤、经传播的信号等。无线通信介质的示例可以包括声学、射频（RF）频谱、红外和其它无线介质。此处使用的术语机器可读的和计算机可读的介质旨在包括存储介质和通信介质。

计算设备 122-1-n 的每一个都可以包括多个数据库。如参考计算设备 122-1 详细示出的，计算设备 122-1 可以被耦合到数据库 124a、124b、126a、126b、128a 和 128b。一般地，每个数据库都可以被安排来存储计算设备 122-1 的各种类型的数据，且具体地，存储应用程序 118a、118b 的数据。例如，在一实施例中，配置数据库 124a、124b 可以存储应用程序 118a、118b 的配置数据。配置数据库 124a 可以存储应用程序 118 的第一版本（例如，应用程序 118a）的配置数据，而配置数据库 124b 可以存储应用程序 118 的第二版本（例如，应用程序 118b）的经转换的配置数据集。例如，在一实施例中，用户简档数据库 126a、126b 可以存储应用程序 118a、118b 的搜索和用户简档数据。用户简档数据库 126a 可以存储应用程序 118 的搜索和用户简档数据，而用户简档数据库 126b 可以存储应用程序 118b 的经转换的配置数据集。例如，在一实施例中，内容数据库 128a、128b 可以存储应用程序 118a、118b 的用户数据或内容数据。内容数据库 128a 可以存储应用程序 118a 的内容数据，而内容数据库 128b 可以存储应用程序 118b 的经转换的内容数据集。可以用计算设备 122-1 来实现其它类型的数据和数据库，且这些实施例不限于该上下文。

在各实施例中，应用程序 118 的每一版本都可以使用配置数据的、搜索和用户简档日期的、和/或内容数据的不同的数据模式。数据模式可以是定义诸如数据类型、数据结构、数据格式、内容、语义、句法等各种数据特征的模型。软件应用程序的不同版本可以使用不同的数据模式。因此，在任何升级操作期间，可能需要在不同的数据模式之间转换耦合到计算设备 122-1 的各数据库所存储的数据。例如，数据库 124a、126a 和 128a 可以存储根据适于与应用程序 118 一起使用的第一数据模式的数据，而数据库 124b、126b 和 128b 可以存储根据适于与应用程序 118b 一起使用的第二数据模式的、分别来自数据库 124a、126a 和 128a 的经转换的数据。各实施例不限于该上下文。

作为示例，内容数据库 128a 可以包括应用程序 118a 的数据集 130。数据集 130 可以包括被称为数据段 132-1-m 的多个数据子集。数据段可以表示任何定义的或离散的数据集或子集。例如，数据段可以对应于 WSS 2.0 版所主存的一给定门户网站或一组分层门户网站的数据子集。视给定实现的需要，数据段

132-1-m 可以具有各不相同的大小或统一的大小。数据集 130 还可以根据适合应用程序 118a 的第一数据模式来存储。

在一般的操作中,网络 100 可被安排成执行计算阵列 140 的一个或多个计算设备 122-1-n 所存储或执行的一个或多个应用程序的逐步升级操作。假定计算设备 122-1-n 的每一个都实现一个或多个应用程序 118,如基于服务器的 web 应用程序。基于服务器的 web 应用程序的一个示例可以包括微软 (MICROSOFT®)公司制造的 WINDOWS® SHAREPOINT® SERVICES (WSS) 2.0 版应用程序。WSS 2.0 版是向例如诸如客户机 180-1-s 等各客户机提供基本团队服务、门户网站和内联网功能的集成的应用程序套件。如数据集 130 所指示的, WSS 2.0 版的用户数据或内容数据可由内容数据库 128a 来存储。虽然作为示例参考 WSS 应用程序或应用程序套件描述了一些实施例,但可以理解,可以使用此处描述的逐步更新技术来升级由单个或多个计算设备执行的任何应用程序。各实施例不限于该上下文。

在某一时刻,假定 WSS 3.0 版的开发完成了,且其已被公布以供一般使用。计算阵列 140 的管理员可能期望将计算阵列 140 的一个或多个计算设备 122-1-n 从 WSS 2.0 版升级到 WSS 3.0 版。为此,管理员可以选择计算设备 122-1-n 中的一个——如计算设备 122-1——经由网络 160 来访问 web 服务器 150 以下载具有 WSS 3.0 版、逐步更新引擎 119、安装程序 120、和附带数据和指令的安装包。另选地,计算设备 122-1 可以经由诸如 CD-ROM 设备等输入设备 114 来加载安装包。计算设备 122-1 可以执行安装包来将包括逐步更新引擎 119 的各安装组件复制的存储器 104。

处理单元 102 可以执行安装程序 120 来安装应用程序 118a 的新版本(例如,应用程序 118b)以供计算设备 122-1 使用。例如,应用程序 118b 可以包括 WSS 3.0 版。安装程序 120 还可以将逐步更新引擎 119 作为应用程序 118b 的集成的或独立的部分来安装。逐步更新引擎 119 可被安排成管理应用程序 118 的升级操作。图形用户界面可以经由诸如监视器或其它类型的显示器等输出设备 116 向管理员提供多个升级选项。这些升级选项可以包括原地升级或逐步升级以及其它选项。原地升级可能是最容易的方式,因为所有的计算设备 122-1-n 都被离线、升级并随后一旦更新操作完成则放回到操作中。然而,原

地升级的代价是整个计算阵列 140 可能离线相当长的一段时间。例如，假定计算阵列 140 是主存 10000 个网站的 web 主存提供商的服务器场。假定升级时间为每个站点一分钟，则其将花大约 10000 分钟或大约一个星期来完成升级操作。作为对比，逐步升级可能在升级操作中引入额外的复杂性，但却允许计算阵列 140 的各部分在升级周期期间维持 WSS 操作，藉此向客户机设备 180-1-j 提供某些受限的功能。

为执行计算阵列 140 的逐步升级，逐步更新引擎 119 最初可以执行对计算设备 122-1 的逐步升级操作。一旦对计算设备 122-1 完成了升级操作，则可以将安装包传递给计算设备 122-2 以执行类似的升级操作。该过程在计算设备 122-n 之中继续，直到计算阵列 140 中的最后一个计算设备被最终升级。

作为计算设备 122-1 的应用程序 118 的升级操作的一部分，逐步更新引擎 119 可以开始转换来自各数据库的各种类型的信息或数据以供与应用程序 118 的第二版本（例如，应用程序 118b）一起使用。例如，逐步更新引擎 119 可以从内容数据库 128a 中检索由数据集 130 所表示的内容数据，并将其从应用程序 118a 所使用的第一数据模式转换到应用程序 118b 所使用的第二数据模式。经转换的数据集 130 可被存储在数据库 128b 中以供应用程序 118b 使用。

逐步更新引擎 119 能以数据段 132-1-m 表示的离散的部分或组块为单位来转换数据集 130。更具体地，逐步更新引擎可以一次一个数据段地转换数据集 130。例如，管理员可以选择一个或多个数据段 132-1-m 以转换到第二数据模式。此外，管理员可以选择数据段 132-1-m 被转换的顺序。假定管理员选择首先转换数据段 132-1，随后是数据段 132-2、132-3，且最后是数据段 132-m。逐步更新引擎 119 可以开始根据所期望的选择和转换顺序将数据段 132-1 转换到第二数据模式。

逐步更新引擎 119 和/或站点管理员可以基于多个因素选择给定一组数据段 132-1-m。例如，可以选择某些数据段 132-1-m 以影响尽可能少的用户。为此，可以在按站点集合的基础上选择，该站点集合是由相同的一组用户和共享许可访问的分层的一组网站或门户网站。在另一示例中，可以基于一组网站的优先级来选择某些数据段 132-1-m。可以首先升级较高优先级网站的数据段，随后升级较低优先级的网站，反之亦然。在又一示例中，可以基于通信量选择

某些数据段 132-1-m。可以首先升级较高通信量网站的数据段，随后升级较低通信量的网站，反之亦然。而在又一示例中，可以基于转换操作所需要的时间量来选择某些数据段 132-1-m。可以选择需要较长转换时间段的数据段在通信量通常较低的晚上或周末期间进行转换操作，而具有较短转换时间段的那些数据段可以选择在通信量通常较高的白天或工作时间进行转换操作。可以理解，这些仅仅是几个示例，且可以使用任何数量的选择准则来选择数据段 132-1-m 被转换到新的数据模式的顺序。各实施例不限于该上下文。

在转换特定的数据段时，逐步更新引擎 119 可以锁定该数据段以阻止对该数据段的任何改变或修改。例如，在转换数据段 132-1 时，逐步更新引擎 119 可以锁定数据段 132-1 以阻止对数据段 132-1 的任何改变。如果在转换操作期间对数据段 132-1 做出改变，则这些改变的数据可能未被转换到 WSS 3.0 版数据模式或可能造成合并冲突。为防止可能的数据丢失或合并冲突，逐步更新引擎 119 可以允许对数据段 132-1 的读特权而没有写特权。另选地，在转换操作期间，逐步更新引擎 119 可以完全阻断对数据段 132-1 的访问。无论如何，只有经常访问正被转换的特定数据段的用户才被转换操作所影响。一旦给定数据段的转换操作完成，逐步更新引擎 119 则可以释放该锁定。

虽然在转换操作期间，逐步更新引擎 119 可以锁定给定数据段，然而，计算设备 122-1 可以允许一个或多个客户机 180-1-s 访问数据集 130 的其它数据段。例如，假定特定门户网站使用数据段 132-2 的数据。客户机 180-1 可以继续使用 WSS 2.0 版应用程序来访问该门户网站，即使逐步更新引擎 119 正将数据段 132-1 转换到新的数据模式。这允许在转换操作期间继续使用 WSS 操作。类似地，一旦数据段 132-1 被转换到新的数据模式，则在其它数据段被转换时，计算设备 122-1 可以允许一个或多个客户机 180-1-s 使用 WSS 3.0 版应用程序和经转换的数据段 132-1 来访问该门户网站。以此方式，基于数据集 130 的给定数据段 132-1-m 的当前转换状态，客户机可以访问 WSS 2.0 版或 WSS 3.0 版的特征。

为减轻客户机 180-1-s 的迁移问题，逐步更新引擎 119 可以基于给定数据段 132-1-m 的转换操作的状态临时地重新定向访问请求。为准备逐步升级操作，逐步更新引擎 119 可以创建新的统一资源定位符（URL）域以供在升级操作期

间使用。例如，假定 WSS 2.0 版位于第一域名 “//domain”（域）。逐步更新引擎 119 可以创建第二域名 “//domain_old”（旧域）以供 WSS Version 2.0 使用。如果客户机 180-1-s 用 URL “//domain/sites/WSS”（WSS 站点域）发送对一网站的访问请求，则逐步更新引擎 119 可以自动地将该访问请求重新定向到 “//domain_old/sites/WSS”（WSS 站点旧域），直到升级操作完成为止。同时，逐步更新引擎 119 可以将 WSS 3.0 版安装到先前 WSS 2.0 版所使用的第一个域 “//domain”。一旦对整个计算阵列 140 完成并确认了的升级操作，逐步更新引擎 119 可以停止重定向操作，藉此向客户机 180-1-s 提供到应用程序 118 的新版本的无缝转换。重定向用户请求以确保可以在升级操作之前和之后浏览原始 URL 域可能减少了对最终用户的影响，因为他们不必获知新的 URL 域。逐步更新引擎 119 与升级操作一致地更新重定向操作，以确保用户总是透明地浏览到正确的且当前的位置。

一旦完成升级操作且数据集 130 被转换到新的数据模式后，逐步更新引擎 119 和/或管理员则可以执行各种有效性和完整性检验以确定升级是否被正确地执行。在确认操作期间，应用程序 118 的先前版本可以保留在临时域名处。如果升级未被正确地执行，逐步更新引擎 119 则可以重新启动重定向操作以将访问请求重定向到临时域名处的应用程序 118 的先前版本。可以持续进行重定向操作直到升级操作完成且被确认满足计算阵列 140 的管理员为止。

逐步更新引擎 119 还可以与各种 API 一起操作以允许其它应用程序结合应用程序 118 执行升级操作。例如，逐步更新引擎 119 可以与各种 API 一起操作以允许第三方应用程序根据对应用程序 118 和数据集 130 的转换操作来转换相对应的数据集。API 可以提供允许开发第三方应用程序以扩展 WSS 2.0 版特征来构建在逐步更新引擎 119 所执行的升级操作上的可扩展性模型。以此方式，第三方应用程序可以将其升级动作协调和构建到主升级过程中。构建在 WSS 2.0 版之上的第三方应用程序的一个示例可以包括 SharePoint Portal Server (SPS) 2003。SPS 2003 可以与 WSS 2.0 版的升级操作相一致地执行其升级操作，藉此进一步减少对于客户机 180-1-s 的停机时间和管理性开销。例如，使用 SPS 2003 的计算设备 122-1 可在 WSS 2.0 版升级到 WSS 3.0 版的同时升级到 Office SharePoint Services(OSS) 2006。

以上各实施例的操作可以参考以下附图和所附示例来进一步描述。一些附图可以包括逻辑流程。虽然此处呈现的这些附图可以包括特定的逻辑流程，但可以理解，逻辑流程仅提供如何实现此处描述的一般功能的示例。此外，除非另外指明，否则给出的逻辑流程不必要以所呈现的顺序执行。另外，给出的逻辑流程可由硬件元素、由处理器执行的软件元素、或其任意组合来实现。各实施例不限于该上下文。

图 2 示出逻辑流程的一个实施例。图 2 示出逻辑流程 200。逻辑流程 200 可以表示此处描述的诸如网络 100、计算设备 112 和/或逐步更新引擎 119 等一个或多个实施例所执行的操作。如逻辑流程 200 所示，可以在框 202 安装应用程序的新版本。可以在框 204 一次一个数据段地将应用程序的先前版本的数据集转换到新版本所使用的数据模式。各实施例不限于该上下文。

例如，在一实施例中，可以将先前版本的第一数据段转换到该数据模式。可以在转换第一数据段的同时提供对先前版本的第二数据段的访问。各实施例不限于该上下文。

例如，在一实施例中，可以将先前版本的第一数据段转换到该数据模式。在转换期间可以阻止对第一数据段的访问。各实施例不限于该上下文。

例如，在一实施例中，可以标识先前版本的域名。可以创建先前版本的临时域名。可以将域名重命名为该临时域名。可以将对该域名的访问请求重定向到该临时域名。各实施例不限于该上下文。

例如，在一实施例中，可以标识先前版本的域名。一旦数据集被转换后，则可以使用新版本的域名。各实施例不限于该上下文。

例如，在一实施例中，其它应用程序可以与使用 API 转换先前版本的数据集相一致地转换相对应的数据集。各实施例不限于该上下文。

例如，在一实施例中，可以选择具有来自一个或多个内容数据库的、web 应用程序所使用的数据的数据段。另选地，可以选择具有来自一个或多个内容数据库的、一组分层的 web 应用程序所使用的数据的数据段。各实施例不限于该上下文。

逐步更新技术可以具有超过传统升级技术的各种优点。例如，逐步更新技术的一个优点是升级操作可以按适合例如定义的停机窗口（诸如在周末器件）

的片断为单位来完成。逐步更新技术还允许从给定应用程序 118 的当前版本的正在工作的环境无缝迁移到应用程序 118 的新的或经更新的版本的正在工作的环境。逐步更新技术还提供一些工具来减少对用户和管理员的工作负担的可能的影响。

所给出的逐步升级技术的性能和效率基于一组关键假设。例如，在一实施例中，逐步升级技术假定应用程序 118 的旧的和新的版本同时运行在同一服务器上。例如，WSS 2.0 版和 WSS 3.0 版两者都可以由计算设备 122-1 执行，而对于 WSS 2.0 版和 WSS 3.0 版存在不同的数据库和虚拟服务器。在某些情况下，虚拟服务器可以运行不同版本的 ASP.Net。还可以有由应用程序 118 的两个版本所服务的多个站点。在升级操作之后，用户可以被自动地定向到适当的域。管理员还将一次将一些站点集合升级到新的数据库和应用程序版本。最后，将保留旧版本数据，且可能将任何站点集合回退或回滚到升级前的应用程序版本和附带数据集。

可以使用各种设计参数和执行约束来构建逐步更新引擎 119 和/或应用程序 118 的性能要求集。例如，在一实施例中，要求集可以包括允许将 WSS 3.0 版安装到当前正运行 WSS 2.0 版的服务器场，而不中断 WSS 2.0 版安装。在另一示例中，要求集可以包括从 WSS 2.0 版数据创建中央管理数据库和配置数据库，并随后创建升级操作所需要的 WSS 3.0 版内容虚拟服务器和内容数据库。而在又一示例中，要求集可以允许管理员更新一个或多个站点集合，而不是一次更新整个系统或数据库。可以将数据传送到执行完全升级的临时数据库，且结果被转移到实况 WSS 3.0 版环境。在升级过程期间，站点集合将变得不可用或只可读，且被升级的站点将具有新的域名。而在又一示例中，要求集可以向管理员提供工具来审阅日志、结果和经升级的 WSS 3.0 版站点，随后选择是否设置从 WSS 2.0 版到 WSS 3.0 版的重定向，使两者都临时可用（例如，WSS 2.0 版将是只读的）并在 WSS 3.0 版上工作，或对所有更新重新激活 WSS 2.0 版并在稍后的日期重新升级。在又一示例中，要求集可以包括继续升级操作直到环境中没有剩余 WSS 2.0 版站点为止。本质上，所有的 WSS 2.0 版站点不是被升级了就是被转移到另一服务器场。在某些情况下，管理员可能需要手动地指示完成升级，且卸载 WSS 2.0 版以完成该过程。可以使用报告来显示什么站点处

于哪一状态中（例如，具有重定向的 WSS 3.0 版、WSS 3.0 版和 WSS 2.0 版、或仅 WSS 2.0 版）。

对于升级诸如 WSS 应用程序等 web 应用程序，逐步升级操作可大致被分为 4 个阶段。第一阶段包括 WSS 2.0 版升级准备和 WSS 3.0 版场创建。安装应用程序 118 的新版本的代码，并将 WSS 2.0 版的全局元素添加到计算阵列 140 中的计算设备 122-1-n 的每一个上的 WSS 3.0 版环境。管理员应当执行任何预升级工作，并使用 WSS 2.0 版站点和问题的预升级报告。

第二阶段包括管理员驱动的 web 应用程序准备。创建新的 web 应用程序虚拟服务器。创建新的内容数据库以在与 WSS 2.0 版内容数据库相同的 SQL 服务器上存储新站点。管理员应当将部署到 WSS 2.0 版虚拟服务器的储存斗的 web 部分添加到 WSS 3.0 版虚拟服务器。

第三阶段包括管理员驱动的站点集合升级。管理员选择一个或多个站点以转移到 WSS 3.0 版。WSS 2.0 版站点离线且不可用，且将所有数据复制到临时数据库。在临时数据库中运行原地升级，随后将结果转移到实况 WSS 3.0 版环境，并使站点在 WSS 3.0 版中可用。一旦升级操作完成，应当使 WSS 2.0 版站点再次可用。

第四阶段包括审阅和重定向。管理员审阅站点。如果站点需要大量工作，或不可升级，则重置到 WSS 2.0 版站点的重定向，且管理员可以任选地创建经升级的 WSS 3.0 版站点的复件。使用网站创建和诸如微软公司制造的 FRONTPAGE®等管理工具，或使用类似的工具来使 WSS 3.0 版复件继续工作，直到用户界面在 WSS 3.0 版环境中正确为止。在该点处，可以重新升级 WSS 2.0 版站点，且使用 FrontPage 或类似的工具手动地合并这些改变。

图 3 示出第二逻辑流程的示例性实施例。图 3 示出可以表示适于逐步升级的安装操作的逻辑流程 300。如逻辑流程 300 所示，在启动逐步升级之前，可以运行预升级扫描工具来为升级操作做准备。预升级扫描工具可以报告供管理员解决的常见问题，提供使用的所有站点定义的列表，更新任何 WSS 列表以便其可被升级等。在执行预升级扫描工具期间，可以检测并在框 302 经由弹出消息，或经由到日志文件的入口向管理员传达升级机会。可以向管理员呈现升级选项的列表，如原地升级、逐步升级、内容迁移升级等。在框 304 可以选择

逐步升级。在框 306，计算设备 122-1 可以检索并加载逐步更新引擎 119 来开始安装应用程序 118 的新版本。在框 308，可以启动后安装配置操作。在该点处，如圈 B 所示，可以将控制传到逻辑流程 400。

图 4 示出第三逻辑流程的示例性实施例。图 4 示出可以表示适于逐步升级的后安装配置操作的逻辑流程 400。如逻辑流程 400 所示，在框 402，执行检验以确定在安装操作期间是否使用了预升级扫描工具。如果未运行预升级扫描工具，则可以生成错误且终止升级操作或将控制传回逻辑流程 300 以重新执行安装操作。如果运行了预升级扫描工具，则在框 404，可以创建具有新应用程序池但具有与 WSS 2.0 版相同的凭证的配置数据库。在框 406，可以执行本地迁移器。本地迁移器的一个示例可以包括执行数据和用于搜索操作的软件的本地升级。随后可以在框 408 启动中央管理主页。

在菱形框 410，做出关于是否为计算阵列 140 中的所有计算设备 122-1-n 都执行了安装操作和后安装配置操作的判定。如果没有为计算阵列 140 中的所有计算设备 122-1-n 执行安装操作和后安装操作，则如圈 A 所示，将控制传到逻辑流程 300。然而，在为所有计算设备 122-1-n 都执行了安装操作和后安装操作时，则如圈 C 所示，将控制传到逻辑流程 500。

图 5 示出第四逻辑流程的示例性实施例。图 5 示出可以表示中央管理和应用程序升级操作的逻辑流程 500。如逻辑流程 500 所示，在框 502，可以选择 web 应用程序来升级。应用程序升级操作可以包括，例如，将 WSS 2.0 版转移到新的 URL 域、在计算设备 122-1 上创建或实例化 WSS 3.0 版 web 应用程序、创建或实例化 WSS 3.0 版的各种数据库以对 WSS 2.0 版所使用的现有数据库制作镜像、将所有的 WSS 2.0 版站点添加到 WSS 3.0 版配置数据库（例如，配置数据库 126b）、为所有站点创建重定向、和/或给定实现所期望的其它升级操作。

继续逻辑流程 500，在框 504，可以定义、修改、更新、或设置各种升级设置。这些设置的示例可以包括 Reghost（登记主存）和 Partner（伙伴）设置。Reghost 设置可以允许管理员在升级操作期间启用登记主存操作。例如，管理员可能期望启用登记主存操作以便在升级操作期间登记主存定制的网页。Partner 设置可以使伙伴和第三方能够在 WSS 逐步升级之上构建升级元素。

一旦设定了升级设置,则在框 506,可以选择和升级一组或多组 SharePoint 站点 (SPSites)。在菱形框 508,可以做出关于是否所有 SPSites 都已被升级到 WSS 3.0 版或回退到 WSS 2.0 版的判定。如果未升级或回退所有 SPSites,则将控制传到框 504。然而,如果已升级或回退了所有 SPSites,则在菱形框 510,可以作出关于对计算设备 122-1 是否已升级了所有应用程序的判定。如果未升级所有的应用程序,则将控制传回到框 502。然而,如果已升级了所有的应用程序,则在框 512 结束升级操作。结束操作的示例可以包括通知管理员和/或用户升级是单向升级、提供确认弹出框、将 WSS 3.0 版场从 WSS 2.0 版场断开连接、清除升级工作数据、和/或给定实现所期望的其它结束操作。

此处阐明了许多具体细节以提供对这些实施例的全面理解。然而,本领域技术人员将理解,没有这些具体细节也可实践这些实施例。在其它实例中,没有详细地描述公知的操作、组件和电路以免得使这些实施例显得晦涩难懂。可以理解,此处公开的具体的结构和功能细节可以是代表性的且不必要限制这些实施例的范围。

还值得注意的是,任何对“一实施例”或“实施例”的引用都意味着结合该实施例所描述的特定的特征、结构、或特性被包括在至少一个实施例中。出现在说明书中各个地方的短语“在一实施例中”不必全都指的是同一实施例。

某些实施例可能使用表述“耦合的”和“连接的”及其派生词来描述。应当理解,这些术语不旨在互为同义词。例如,某些实施例可能使用术语“连接的”来描述以指示两个或更多元素互相有直接的物理的或电的接触。在另一示例中,某些实施例可能使用术语“耦合的”来描述以指示两个或更多元素有直接的物理的或电的接触。然而,术语“耦合的”还可以意味着两个或更多元素互相不直接接触,而仍互相合作或交互。各实施例不限于该上下文。

某些实施例可以例如使用可以存储如果被执行则使得机器执行根据这些实施例的方法和/或操作的指令或一组指令的机器可读介质或物品来实现。例如,这种机器可以包括任何合适的处理平台、计算平台、计算设备、计算系统、处理系统、计算机、处理器等,且可以使用硬件和/或软件的任意合适的组合来实现。例如,机器可读介质或物品可以包括任何合适类型的存储器单元、存储器设备、存储器物品、存储器介质、存储设备、存储物品、存储介质和或存储

单元，例如，存储器、可移动或不可移动介质、可擦除或不可擦除介质、可写或可重写介质、数字或模拟介质、硬盘、软盘、压缩盘只读存储器（CD-ROM）、可记录压缩盘（CD-R）、可重写压缩盘（CD-RW）、光盘、磁性介质、磁光盘、可移动存储器卡或盘、各种类型的数字多功能盘、磁带、磁盒等。

尽管用对结构特征和/或方法动作专用的语言描述了本主题，但可以理解，所附权利要求书中定义的主题不必限于上述具体特征或动作。相反，上述具体特征和动作是作为实现权利要求的示例形式公开的。

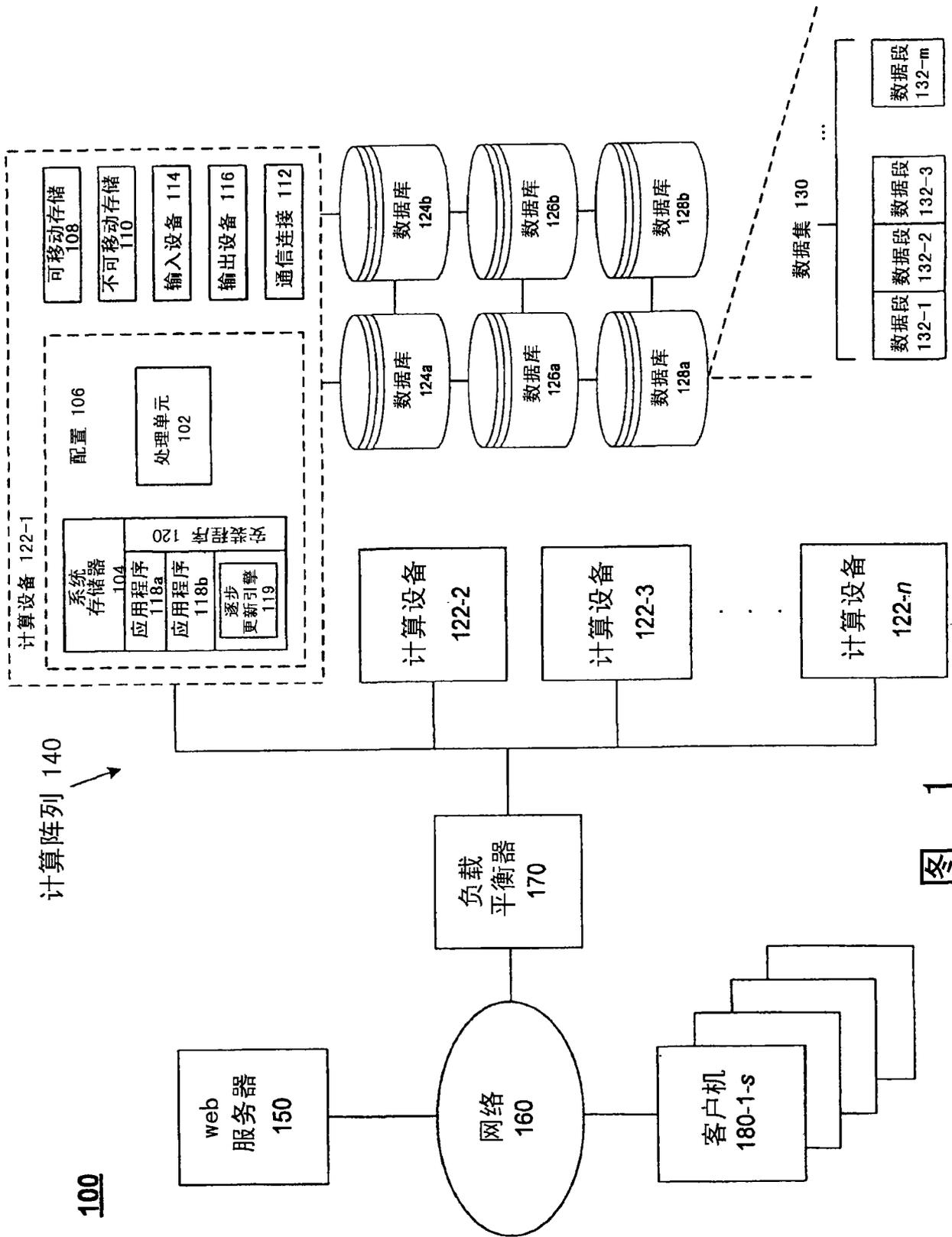


图 1

200

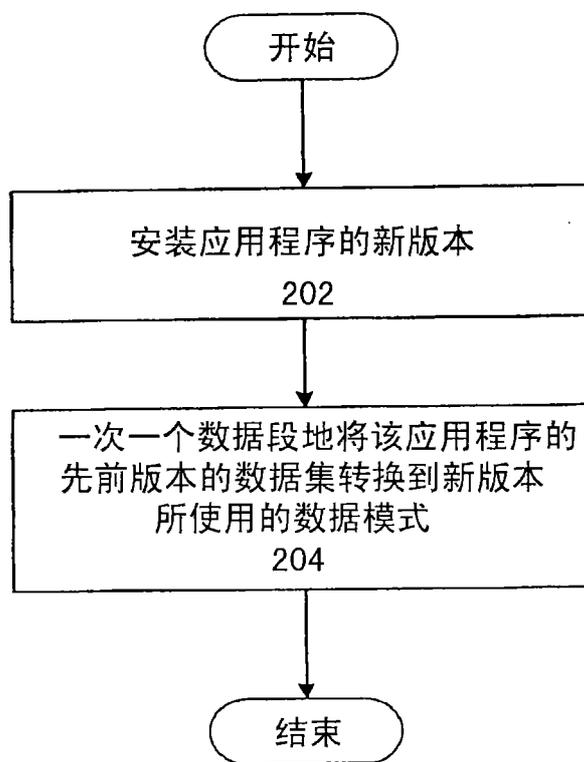


图 2

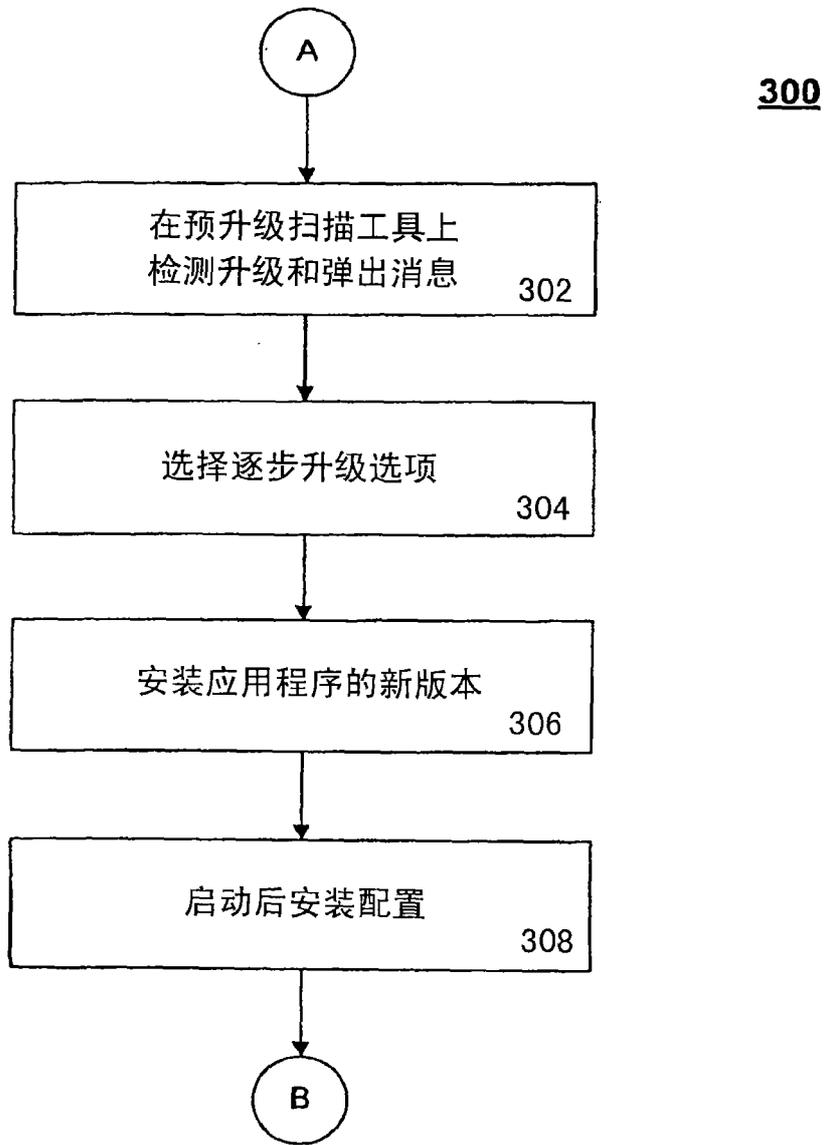


图 3

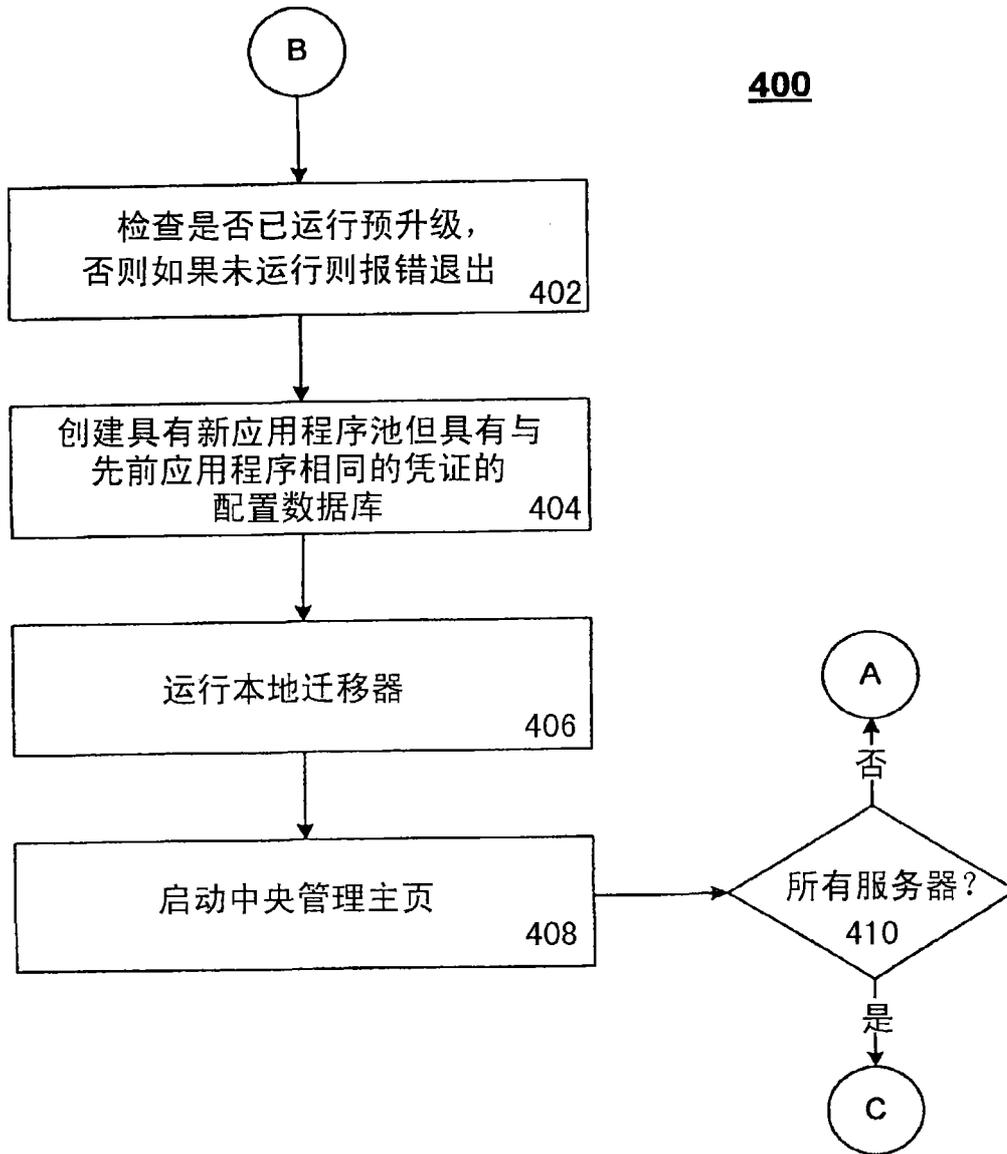


图 4

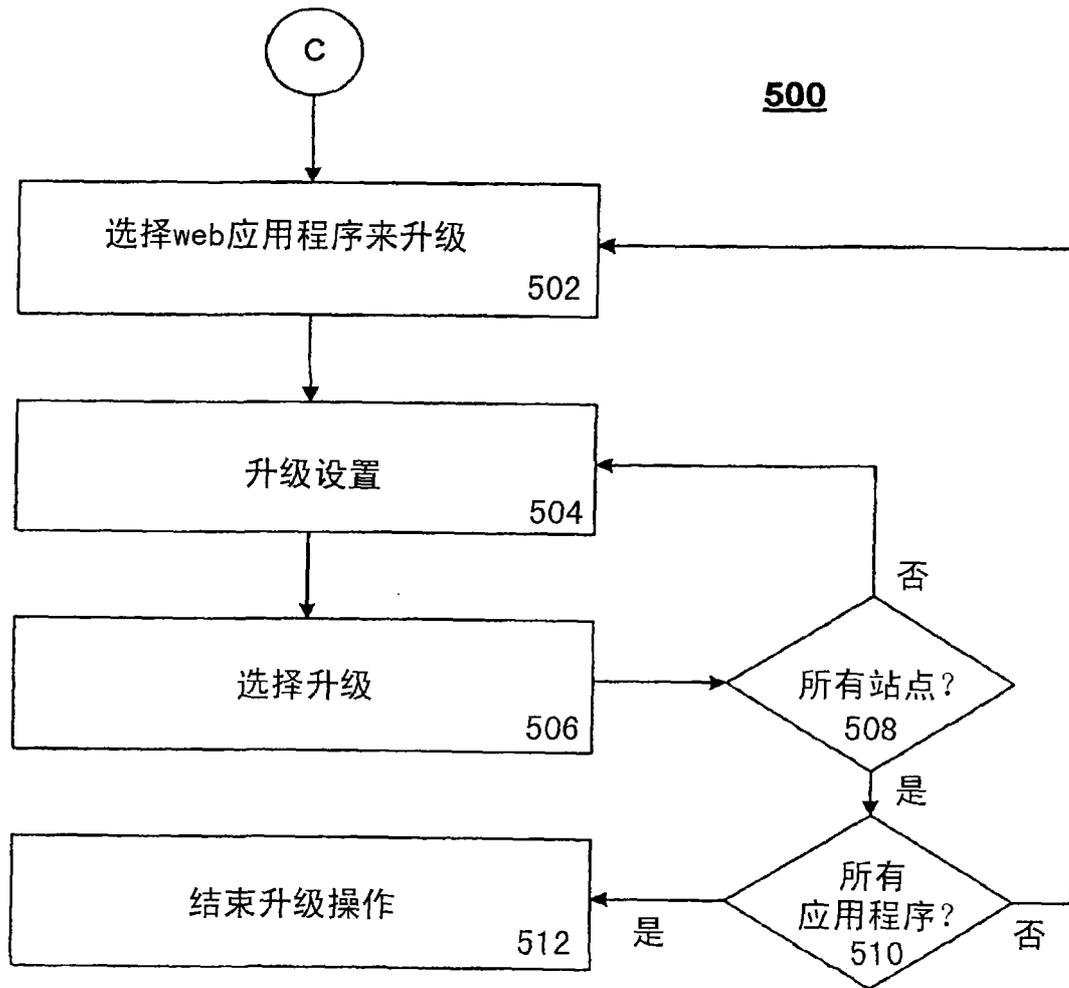


图 5