



(12) 发明专利

(10) 授权公告号 CN 102567042 B

(45) 授权公告日 2015. 04. 15

(21) 申请号 201110360028. 2

(22) 申请日 2011. 11. 15

(30) 优先权数据

10194866. 9 2010. 12. 14 EP

(73) 专利权人 国际商业机器公司

地址 美国纽约

(72) 发明人 C·马里内利 J·方迪格尼耶

M·V·斯图克尔伯格 J·G·茹内

D·克拉克 L·加尔斯-埃里斯

(74) 专利代理机构 中国国际贸易促进委员会专

利商标事务所 11038

代理人 杜娟

(51) Int. Cl.

G06F 9/445(2006. 01)

(56) 对比文件

US 2010205594 A1, 2010. 08. 12,

US 2008010639 A1, 2008. 01. 10,

US 2004088367 A1, 2004. 05. 06,

CN 101256527 A, 2008. 09. 03,

审查员 田冰

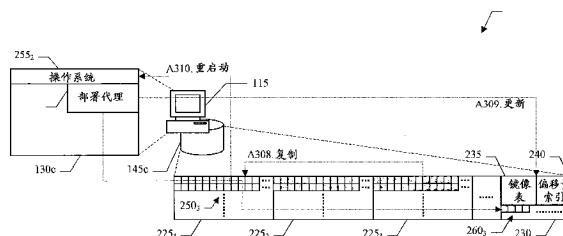
权利要求书4页 说明书12页 附图18页

(54) 发明名称

利用引导块重定位来管理多个软件镜像的方法和系统

(57) 摘要

本发明的实施例涉及利用引导块重定位来管理多个软件镜像的方法和系统。用于在数据处理实体中管理多个软件镜像的方法包括步骤:将每个软件镜像的至少一部分存储块存储到大容量存储器的对应镜像部分中,每个存储块被存储到具有等于对应的镜像地址加上大容量存储器内该镜像部分的偏移量的存储地址的存储位置中,选择存储在当前镜像中的当前软件镜像,将存储在当前软件镜像的引导位置中的存储块重定位到大容量存储器的重定位部分,将当前软件镜像的引导块复制到对应的引导位置中,从当前软件镜像的对应引导位置中的引导块启动该数据处理实体,由此加载该访问功能,以及服务每一个通过该访问功能来访问当前软件镜像的所选择的存储块的请求。



1. 一种用于在数据处理实体 (115) 中管理多个软件镜像的方法, 该数据处理实体 (115) 包括具有多个存储位置的大容量存储器 (145c), 每个存储位置在大容量存储器中具有对应的存储地址, 每个软件镜像包括多个存储块, 每个存储块在软件镜像中具有对应的镜像地址, 其中该方法包括下列步骤:

将每个软件镜像的至少一部分存储块存储到大容量存储器的对应镜像部分 (225_i) 中, 每个存储块被存储到具有等于对应的镜像地址加上大容量存储器内该镜像部分的偏移量的存储地址的存储位置中,

选择存储在当前镜像中的当前软件镜像,

将存储在当前软件镜像的引导位置中的存储块重定位 (A307) 到大容量存储器的重定位部分 (230), 当前软件镜像的引导位置是具有等于当前软件镜像的引导块 (250_i) 的镜像地址的存储地址的存储位置, 所述引导块 (250_i) 包括启动数据处理实体直到加载适于访问当前软件镜像的访问功能所需的存储块,

将当前软件镜像的引导块复制 (A308) 到对应的引导位置中,

从当前软件镜像的对应引导位置中的引导块启动 (A310) 该数据处理实体, 由此加载该访问功能, 以及

通过该访问功能来服务每一个访问当前软件镜像的所选择的存储块的请求, 该访问功能在具有等于对应的镜像地址加上当前镜像部分的偏移量的存储地址的存储位置中访问所选择的存储块。

2. 根据权利要求 1 的方法, 还包括步骤:

在将存储在当前软件镜像的引导位置中的存储块重定位 (A307) 到重定位部分之前, 把从前一个软件镜像的引导位置重定位到重定位部分的存储块恢复 (A306) 到前一个软件镜像的引导位置中。

3. 根据权利要求 1 或 2 的方法, 其中该镜像部分包括具有零偏移量的初始镜像部分, 所有镜像部分的引导位置被包括在该初始镜像部分中, 而且其中, 仅当当前镜像部分不同于初始镜像部分时, 才执行重定位 (A307) 存储在当前软件镜像的引导位置中的存储块和复制 (A308) 当前软件镜像的引导块的步骤。

4. 根据权利要求 1 或 2 中任一项的方法, 还包括步骤:

选择存储在源镜像部分中的源软件镜像作为快照, 以及

将源软件镜像的每个存储块复制 (A302) 到一个空闲的镜像部分中, 当源软件镜像的存储块被重定位在重定位部分中时从重定位部分复制源软件镜像的存储块, 否则从源镜像部分复制源软件镜像的存储块。

5. 根据权利要求 1 或 2 中任一项的方法, 还包括步骤:

选择存储在要删除的过时镜像部分中的过时软件镜像, 以及

将该过时镜像部分设置为空闲。

6. 根据权利要求 1 或 2 中任一项的方法, 其中每个软件镜像的所有存储块存储在对应的镜像部分中。

7. 根据权利要求 1 或 2 中任一项的方法, 其中从当前软件镜像的引导块启动数据处理实体的步骤包括:

根据当前镜像部分的大小, 通过访问功能来调整当前软件镜像的大小。

8. 根据权利要求 1 或 2 中任一项的方法,还包括步骤:

请求将新软件镜像从外部源部署到数据处理实体的新镜像部分上,

将新软件镜像的引导块从外部源下载到数据处理实体上,

将存储在新软件镜像的引导位置中的存储块重定位 (A210) 到重定位部分,

将新软件镜像的引导块复制到对应的引导位置中,

从新软件镜像的对应引导位置中的引导块启动 (A214) 数据处理实体,由此加载访问功能,以及

通过该访问功能来服务每一个访问新软件镜像的另外所选择的存储块的请求,该访问功能响应于大容量存储器中缺少该另外所选择的存储块而从外部源下载 (A216a) 该另外所选择的存储块并将该另外所选择的存储块存储 (A217) 到具有等于对应的镜像地址加上新镜像部分的偏移量的存储地址的存储位置中,或者从具有等于对应的镜像地址加上新镜像部分的偏移量的存储地址的存储位置检索 (A216b) 该另外所选择的存储块。

9. 根据权利要求 8 的方法,还包括步骤:

在重定位 (A210) 存储在新软件镜像的引导位置中的存储块之前,把从当前软件镜像的引导位置重定位到重定位部分的存储块恢复 (A209) 到当前软件镜像的引导位置中。

10. 根据权利要求 8 的方法,还包括步骤:

将新软件镜像的引导块复制到具有等于对应的镜像地址加上新镜像部分的偏移量的存储地址的存储位置中。

11. 根据权利要求 8 的方法,其中对于以更新模式访问另外所选择的存储块的每一个请求 (A215),该方法还包括步骤:

更新具有等于对应的镜像地址加上新镜像部分的偏移量的存储地址的存储位置中的另外所选择的存储块。

12. 根据权利要求 8 的方法,还包括步骤:

监视数据处理实体的工作负载,以及

响应于工作负载下降到阈值之下,从外部源下载一组未被存储在大容量存储器中的新的存储块,并将新的存储块存储到具有等于对应的镜像地址加上新镜像部分的偏移量的存储地址的存储位置中。

13. 根据权利要求 1 或 2 中任一项的方法,还包括通过下述操作来准备每个软件镜像的步骤:

从软件镜像模拟 (A404) 辅助数据处理实体的引导序列,直到加载访问功能,在引导序列期间被访问的软件镜像的每一个存储块被跟踪,以及

根据被跟踪的存储块,识别 (A405) 软件镜像的引导块。

14. 一种用于在数据处理实体 (115) 中管理多个软件镜像的系统,该数据处理实体 (115) 包括具有多个存储位置的大容量存储器 (145c),每个存储位置在大容量存储器中具有对应的存储地址,每个软件镜像包括多个存储块,每个存储块在软件镜像中具有对应的镜像地址,其中该系统包括下列模块:

配置为将每个软件镜像的至少一部分存储块存储到大容量存储器的对应镜像部分 (225_i) 中,每个存储块被存储到具有等于对应的镜像地址加上大容量存储器内该镜像部分的偏移量的存储地址的存储位置中的模块,

配置为选择存储在当前镜像中的当前软件镜像的模块,

配置为将存储在当前软件镜像的引导位置中的存储块重定位 (A307) 到大容量存储器的重定位部分 (230) 的模块, 当前软件镜像的引导位置是具有等于当前软件镜像的引导块 (250_i) 的镜像地址的存储地址的存储位置, 所述引导块 (250_i) 包括启动数据处理实体直到加载适于访问当前软件镜像的访问功能所需的存储块,

配置为将当前软件镜像的引导块复制 (A308) 到对应的引导位置中的模块,

配置为从当前软件镜像的对应引导位置中的引导块启动 (A310) 该数据处理实体, 由此加载该访问功能的模块, 以及

配置为通过该访问功能来服务每一个访问当前软件镜像的所选择的存储块的请求的模块, 该访问功能在具有等于对应的镜像地址加上当前镜像部分的偏移量的存储地址的存储位置中访问所选择的存储块。

15. 根据权利要求 14 的系统, 还包括:

配置为在将存储在当前软件镜像的引导位置中的存储块重定位 (A307) 到重定位部分之前, 把从前一个软件镜像的引导位置重定位到重定位部分的存储块恢复 (A306) 到前一个软件镜像的引导位置中的模块。

16. 根据权利要求 14 或 15 的系统, 其中该镜像部分包括具有零偏移量的初始镜像部分, 所有镜像部分的引导位置被包括在该初始镜像部分中, 而且其中, 仅当当前镜像部分不同于初始镜像部分时, 才执行重定位 (A307) 存储在当前软件镜像的引导位置中的存储块和复制 (A308) 当前软件镜像的引导块的模块。

17. 根据权利要求 14 或 15 中任一项的系统, 还包括:

配置为选择存储在源镜像部分中的源软件镜像作为快照的模块, 以及

配置为将源软件镜像的每个存储块复制 (A302) 到一个空闲的镜像部分中, 当源软件镜像的存储块被重定位在重定位部分中时从重定位部分复制源软件镜像的存储块, 否则从源镜像部分复制源软件镜像的存储块的模块。

18. 根据权利要求 14 或 15 中任一项的系统, 还包括:

配置为选择存储在要删除的过时镜像部分中的过时软件镜像的模块, 以及

配置为将该过时镜像部分设置为空闲的模块。

19. 根据权利要求 14 或 15 中任一项的系统, 其中每个软件镜像的所有存储块存储在对应的镜像部分中。

20. 根据权利要求 14 或 15 中任一项的系统, 其中从当前软件镜像的引导块启动数据处理实体的模块包括:

配置为根据当前镜像部分的大小, 通过访问功能来调整当前软件镜像的大小的模块。

21. 根据权利要求 14 或 15 中任一项的系统, 还包括:

配置为请求将新软件镜像从外部源部署到数据处理实体的新镜像部分上的模块,

配置为将新软件镜像的引导块从外部源下载到数据处理实体上的模块,

配置为将存储在新软件镜像的引导位置中的存储块重定位 (A210) 到重定位部分的模块,

配置为将新软件镜像的引导块复制到对应的引导位置中的模块,

配置为从新软件镜像的对应引导位置中的引导块启动 (A214) 数据处理实体, 由此加

载访问功能的模块,以及

配置为通过该访问功能来服务每一个访问新软件镜像的另外所选择的存储块的请求,该访问功能响应于大容量存储器中缺少该另外所选择的存储块而从外部源下载 (A216a) 该另外所选择的存储块并将该另外所选择的存储块存储 (A217) 到具有等于对应的镜像地址加上新镜像部分的偏移量的存储地址的存储位置中,或者从具有等于对应的镜像地址加上新镜像部分的偏移量的存储地址的存储位置检索 (A216b) 该另外所选择的存储块的模块。

22. 根据权利要求 21 的系统,还包括:

配置为在重定位 (A210) 存储在新软件镜像的引导位置中的存储块之前,把从当前软件镜像的引导位置重定位到重定位部分的存储块恢复 (A209) 到当前软件镜像的引导位置中的模块。

23. 根据权利要求 21 的系统,还包括:

配置为将新软件镜像的引导块复制到具有等于对应的镜像地址加上新镜像部分的偏移量的存储地址的存储位置中的模块。

24. 根据权利要求 21 的系统,其中对于以更新模式访问另外所选择的存储块的每一个请求 (A215),该系统还包括:

配置为更新具有等于对应的镜像地址加上新镜像部分的偏移量的存储地址的存储位置中的另外所选择的存储块的模块。

25. 根据权利要求 21 的系统,还包括:

配置为监视数据处理实体的工作负载的模块,以及

配置为响应于工作负载下降到阈值之下,从外部源下载一组未被存储在大容量存储器中的新的存储块,并将新的存储块存储到具有等于对应的镜像地址加上新镜像部分的偏移量的存储地址的存储位置中的模块。

26. 根据权利要求 14 或 15 中任一项的系统,还包括通过下述操作来准备每个软件镜像的模块:

配置为从软件镜像模拟 (A404) 辅助数据处理实体的引导序列,直到加载访问功能,在引导序列期间被访问的软件镜像的每一个存储块被跟踪的模块,以及

配置为根据被跟踪的存储块,识别 (A405) 软件镜像的引导块的模块。

利用引导块重定位来管理多个软件镜像的方法和系统

技术领域

[0001] 根据本发明一个或多个实施例的方案涉及数据处理领域。更具体地,本方案涉及软件镜像 (software image) 的管理。

背景技术

[0002] 一般来说,软件镜像包括驻留于计算机上的软件模块(例如,其操作系统、应用程序和/或数据)的结构。

[0003] 一些情况下,可能需要在同一计算机上具有多个可用的操作系统。例如,这可能有助于运行需要不同操作系统的程序,或有助于测试新的操作系统或者其新版本。

[0004] 出于这个目的,能够利用多引导 (multi-boot) 技术。多引导技术允许在计算机上安装多个操作系统,当计算机启动时,能够选择启动哪一个操作系统。通过将计算机的硬盘分割成多个分区来实现所需要的结果,每个分区定义一个逻辑盘用于存储对应的操作系统。计算机从主分区启动,主分区包括一个引导装载程序 (boot loader),其允许选择要启动的实际操作系统;可替换地,可以给引导分区提供主引导装载程序,其可简单地用于选择所需要的操作系统;然后为启动所选择的操作系统,主引导装载程序调用该所选择的操作系统的第二引导装载程序。

[0005] 然而,多引导技术很严格,因为硬盘的分区被提前定义。无论如何,一旦所选择的操作系统已经启动,它就完全控制整个硬盘;因此,所选择的操作系统同样可以访问其它的分区(有毁坏它们的风险)。

[0006] 可选择地,通过开发虚拟化技术也可以实现同样的结果。在这种情形下,管理程序被安装在计算机上。该管理程序实现一个虚拟化层,其仿真多个虚拟机,每个虚拟机由一个给出物理计算机的外观的抽象环境(它被虚拟机唯一控制)构成。这样,不同的操作系统可以独立(甚至同时)地运行在对应的虚拟机上。

[0007] 然而,虚拟化技术需要安装复杂的基础设置来管理虚拟机。而且,这将导致计算机性能的下降(因为操作系统不再自然地运行在计算机上)。

[0008] US-A-2009/0193245(此处引用包含了其全部公开内容)也公开了一种将多引导计算机转换成虚拟机的方法。出于这个目的,配置计算机的引导记录,以加载管理虚拟机的宿主操作系统;一个转换器从对应的引导镜像生成虚拟机,转换器检测并应用相同的配置,并解决任何可能由虚拟机并发运行所导致的冲突。如上,这需要安装一个虚拟化层(由宿主操作系统实现);而且,操作系统同样在一个虚拟化的环境(由宿主操作系统提供)中运行,伴随着相应的性能下降。

[0009] 另一个常见的需求是对计算机的软件镜像做出备份副本 (backup copy)。例如,在发生故障的情况下,这有助于恢复计算机的内容。

[0010] 出于这个目的,可以获取软件镜像的快照(即:其在一特定时间点处于一致状态的备份副本);可以将快照保存在备份盘上或备份服务器上。这样,可以通过从备份盘或备份服务器将快照重新安装到计算机上来恢复快照。然而,恢复快照的过程非常缓慢。另外,

在使用备份服务器的情况下,这将涉及网络资源的高消耗;而且,需要与备份服务器的网络连接,以从该备份服务器恢复快照。可选择地,可以从备份服务器上的快照远程启动计算机。然而,在这种情形下,计算机必须总与备份服务器连接以用于其操作;无论如何,通过网络的计算机操作会导致其性能的下降。

发明内容

[0011] 概括地说,根据本发明一个或多个实施例的方案是基于通过重定位其引导块来管理多个软件镜像的构思。

[0012] 具体地,独立权利要求阐述了根据本发明特定实施例的方案的一个或多个方面,从属权利要求阐述了相同方案的有利特征,此处其措词通过引用被逐字地包括进来(关于根据本发明实施例的方案的一个特定方面提供的任何有利的特征对其每一个其它方面在细节上做必要修改后适用)。

[0013] 更具体地,根据本发明实施例的方案的一个方面提供了一种用于管理数据处理实体(例如,一台独立的计算机)中多个软件镜像的方法。该数据处理实体包括一个大容量存储器(mass memory),该大容量存储器具有多个存储位置(memory location);每个存储位置在大容量存储器内具有一个对应的存储地址。进而,每个软件镜像包括多个存储块(memory block);每个存储块在软件镜像内具有一个对应的镜像地址。该方法包括下列步骤。将每个软件镜像的存储块(或其至少一部分)存储到大容量存储器的一个对应的镜像部分中;特别地,每个存储块被存储到具有等于对应的镜像地址加上大容量存储器内镜像部分的偏移量(offset)的存储地址的存储位置中。选择存储在当前镜像部分中的当前软件镜像(例如,通过切换至先前保存的快照)。将存储在当前软件镜像的引导位置(boot location)中的存储块重定位到大容量存储器的重定位部分;当前软件镜像的引导位置是具有等于当前软件镜像的引导块的镜像地址的存储地址的存储位置,该引导块包括启动数据处理实体直到加载访问功能(适用于访问当前软件镜像)所需的当前软件镜像的存储块。将当前软件镜像的引导块复制到对应的引导位置中。从当前软件镜像的在对应引导位置中的引导块启动数据处理实体,由此加载访问功能。访问功能处理每一个访问当前软件镜像的所选择的存储块的请求;访问功能在具有等于对应的镜像地址加上当前镜像部分的偏移量的存储地址的存储位置中访问所选择的存储块。

[0014] 根据本发明实施例的方案的另一方面提供了一种计算机程序,其包括代码装置,用于当在数据处理系统上执行该计算机程序时,使数据处理系统执行本方法的步骤;根据本发明实施例的方案的一方面提供了一种计算机程序产品,其包括一个非瞬时的计算机可读介质,该介质包含计算机程序,该计算机程序包括被直接装载进数据处理系统的工作存储器中的代码装置,由此配置数据处理系统以执行相同的方法。

[0015] 根据本发明实施例的方案的另一方面提供了一种包括用于执行所述方法步骤的装置的系统。

[0016] 附图简要说明

[0017] 根据本发明的一个或多个实施例的方案,除了更多的特征和其优点之外,将参考下面的具体描述而被最好地理解,下面的描述纯粹是通过非限制性的指示的方式来给出,其将与附图(其中,为了简便的目的,对应的元件用相同或相似的标记来表示,且不重复它

们的说明,每一个实体的名称通常用于表示其类型和属性(例如它的值、内容和表示)一起阅读。具体地:

[0018] 图 1 显示了一个数据处理系统的示意性框图,其中可应用根据本发明实施例的方案,

[0019] 图 2A-图 2K 显示了一个表示可以用来实现根据本发明实施例的部署过程的主要软件组件的角色的协作图,

[0020] 图 3A-图 3E 显示了一个表示可以用来实现根据本发明实施例的快照过程的主要软件组件的角色的协作图,以及

[0021] 图 4 显示了一个表示可以用来实现根据本发明实施例的一个准备过程的主要软件组件的角色的协作图。

具体实施方式

[0022] 具体参考图 1,显示了一个数据处理系统(或简称为系统)100 的示意性框图,其中可适用根据本发明的一实施例的方案。系统 100 具有分布式体系结构,其基于网络 105—例如,局域网(LAN)。多个计算机通过网络 105 彼此相连。尤其是,服务器计算机 110 控制将软件镜像部署到客户端计算机 115 上—出于简化的目的图中只显示了两个;每个软件镜像是包括一个或多个软件模块(例如,操作系统、应用程序、和/或数据)的结构。

[0023] 系统 100 的一个通用(服务器或客户端)计算机由若干个单元形成,这些单元并行连接到系统总线 120(其具有根据系统 100 中计算机的实际功能适当扩展的结构)。具体地,一个或多个微处理器(μ P)125 控制计算机的操作;RAM130 被微处理器 125 用作工作存储器,ROM135 存储计算机的基本代码。若干个外围单元聚集在局部总线 140 周围(借助各自接口)。尤其是,大容量存储器包括一个或多个硬盘 145 和用于读取光盘 155(例如,DVD 或 CD)的驱动器 150。而且,计算机包括输入单元 160(例如,键盘和鼠标)和输出单元 165(例如,监视器和打印机)。使用适配器 170 连接计算机至网络 105。桥接单元 175 将系统总线 120 和局部总线 140 相连接。每个微处理器 125 和桥接单元 175 能够作为主代理操作,其请求访问系统总线 120 以便传送信息。仲裁器 180 管理对系统总线 120 的互斥访问的授权。

[0024] 图 2A-图 2K 显示了一协作图,其表示可以用于实现根据本发明实施例的部署过程的主要软件组件的角色。具体地,图形描述了该系统的静态结构(借助相应的组件,整体用标记 200 表示)和它的动态行为(借助一系列交换的消息,每个消息表示相应的动作,用前面具有符号“A”的进展序列号表示)。

[0025] 从图 2A 开始,服务器计算机 110(它的工作存储器与大容量存储器分别用标记 130s 和 145s 表示)运行一个部署管理器 205—例如 IBM 公司的 IBM Tivoli 操作系统部署供应管理器(或用于 OSD 的 TPM)的 IBM Tivoli 镜像供应管理器(或 TPMfI)(IBM 和 Tivoli 是 IBM 公司的商标);部署管理器 205 被用于自动部署软件镜像 210_i ($i = 1 \dots N$) 至系统的客户端计算机上。软件镜像 210_i 存储在相应的贮藏库(repository)中。每个软件镜像 210_i 包括一组存储块,存储块在软件镜像 210_i 内具有对应的地址(称之为镜像地址);存储块可以包括任何类型的信息(例如,一个或多个扇区、文件、库、目录及上述的组合或一部分,或者是与操作系统或者是与应用程序相关)。

[0026] 尤其是,每当一个新的软件镜像 210_i (例如,软件镜像 210_1) 要被部署到特定的客户端计算机 115 上(它的工作存储器与大容量存储器分别用标记 130c 和 145c 表示),系统的操作员 212 就通过部署管理器 205 选择这些客户端计算机 115 和新软件镜像 210_i —例如,通过与在图中未示出的另一个客户端计算机上运行的浏览器相连(动作“A201. 选择”)。作为响应,部署管理器 205 开启客户端计算机 115。因而,假设客户端计算机 115 不具有任何起作用的操作系统,它通过网络(图中未示出)启动。具体地,存储在客户端计算机 115 的固件中并于开机时执行的引导装载程序—例如,基本输入输出系统(BIOS)—找不到任何可启动的设备,则起动(launch)一个网络引导装载程序—例如,嵌入到网络适配器中的预启动执行环境(Preboot Execution Environment, PXE)。网络引导装载程序利用一种动态地址服务—例如,基于动态主机配置协议(DHCP)—来从服务器计算机 110(充当 DHCP 服务器)获取客户端计算机 115 的动态地址,服务器计算机 110 还提供了一个网络引导程序(network bootstrap program)的地址,该网络引导程序被下载至客户端计算机 115 的 RAM 盘中—即,被当作大容量存储器的其工作存储器 130c 的一部分—然后被起动。网络引导程序—例如,Windows 预安装环境(WinPE),Windows 是微软公司的商标—提供了一个最小操作系统 215,它包括一个部署代理 220 用于与部署管理器 205 交互(动作“A202. 网络启动”)。

[0027] 部署代理 220 通过将大容量存储器 145c 在逻辑上分割成多个镜像部分 225_j ($j = 0 \dots M$ —例如, $M = 2-10$) 和服务部分 230 来对其进行初始化。镜像部分 225_j (每一个用于存储一个软件镜像)都具有相同的大小(至少与可以在其上安装的一个软件镜像相等),并且从大容量存储器 145c 的开始处被连续排列,而服务部分 230 (用于存储服务信息)较小且被安排在大容量存储器 145c 的结束处。尤其是,服务部分 230 存储镜像表 235,它包括镜像部分 225_j 的状态信息。例如,对于每个镜像部分 225_j ,镜像表 235 包括一条具有指示其可用性—例如,空闲或被使用—的标志的记录(在开始时所有标志都被取消断言(deasserted),表明所有镜像部分 225_j 是空闲的),以及一个指示其在大容量存储器 145c 中位置的从其开始处计算的偏移量(根据镜像部分 225_j 的大小来计算);服务部分 230 还存储一个偏移量索引 240,其表示当前激活的镜像部分 225_j 的偏移量(动作“A203. 初始化”)。

[0028] 部署代理 220 然后从服务器计算机 110 下载新软件镜像 210_i 的一组引导块;为这个目的,部署代理 220 充当一个远程访问启动器,它与服务器计算机 110 的远程访问服务器 245 交互—例如,基于互联网小型计算机系统接口(iSCSI)协议。所述引导块包括启动新软件镜像 210_i 的引导序列(boot sequence)直到装载部署代理 220 所需的存储块;例如,在微软 Windows 中,所述引导块包括(除了部署代理 220 之外)主引导记录(Master Boot Record, MBR)、引导扇区、bootmgr.exe 文件、boot\bcd 文件、系统注册表、winload.exe 文件和系统注册表中规定的驱动文件(动作“A204. 下载”)。

[0029] 这时候,部署代理 220 识别初始镜像部分 225_1 为空闲的,以接收新软件镜像 210_i (因为它是镜像表 235 中具有被取消断言的对应标志的第一个)。结果,部署代理 220 相应地更新镜像表 235 (通过断言(assert) 镜像部分 225_1 的标志,来指示它被使用)和偏移量索引 240 (通过将它设置到镜像表 235 中所指示的新镜像部分 225_1 的偏移量,此时为零,来指示它是当前的一个);而且,部署代理 220 将引导块的镜像地址保存到镜像表 235 中与

当前镜像部分 225₁ 相关联的记录中。部署代理 220 还在当前镜像部分 225₁ 中（在其末端，在它的被保留用于存储新软件镜像 210₁ 的部分之后）创建一个位置图（图中未示出）。对于软件镜像 210₁ 的每一个存储块，该位置图包括一个指示它在大容量存储器 145c 中的可用性的标志—开始时所有标志被取消断言，表明大容量存储器 145c 中还没有可用的存储块（动作“A212. 配置”—具有下面将描述的与跳过的序列号相对应的动作）。部署代理 220 然后将新软件镜像 210₁ 的引导块存储到当前镜像部分 225₁ 中，也就是，存储到大容量存储器 145c 的存储位置中，该存储位置在大容量存储器 145c 中的地址（称为存储地址）等于对应的镜像地址加上当前镜像部分 225₁ 的偏移量—此处它们用灰色显示，并用标记 250₁ 表示；同时，位置表中的对应标志被断言（以指示该引导块现在在大容量存储器 145c 中可用）。由于在这种情况下当前镜像部分 225₁ 的偏移量为零，引导块 250₁ 被准确设置在大容量存储器 145c 中期望在引导序列期间能够找到它们的地方（动作“A213. 存储”）。这时候，部署代理 220 关闭然后开启客户端计算机 115。

[0030] 因此，如图 2B 中所示，客户端计算机 115 正常地从大容量存储器 145c 启动。事实上，客户端计算机 115 的在开机时执行的引导装载程序现在将大容量存储器 145c 识别为一个可启动的设备，以便它从它的引导块 250₁ 本地启动。这样，与引导块 250₁ 相对应的新软件镜像 210₁ 的实际操作系统的一部分（用标记 255₁ 表示）和部署代理 220 被装载到工作存储器 130c 中；例如，在微软 Windows 中，BIOS 装载 MBR，MBR 装载引导扇区，引导扇区查找并启动 bootmgr.exe 文件，bootmgr.exe 文件查找并读取 boot\bcd 文件以确定存储位置，然后装载系统注册表、winload.exe 文件和注册表中规定的驱动文件，并且 winload.exe 启动部署代理 220。当新软件镜像 210₁ 的大小与当前镜像部分 225₁ 的大小不相同（例如，比它小）时，部署代理 220 还调整其大小，以便占用整个的当前镜像部分 225₁；此操作非常快速，因为它只需要更新操作系统 255₁ 的文件系统结构，图中未示出（动作“A214. 本地启动”）。在客户端计算机 115 的操作期间，每一个访问新软件镜像 210₁ 的所选择的存储块的请求现在在被部署代理 220 的一个流式驱动器（streaming driver）服务（它取代（override）操作系统 255₁ 的一个标准文件系统驱动器，图中未示出）。

[0031] 具体地，文件系统驱动器接收一个访问（如，读取）所选择的存储块的请求—例如，从图中未示出的应用程序（动作“A215. 访问请求”）。该请求被传给部署代理 220，部署代理 220 检验所选择的存储块在当前镜像部分 225₁ 中是否可用（如在其位置图中所指示的）。当所选择的存储块在当前镜像部分 225₁ 中不可用（即，位置图中其标志被取消断言）时，部署代理 220 将该请求传给操作系统 255₁ 的一个远程访问驱动器（在所讨论示例中充当 iSCSI 启动器），图中未示出；远程访问驱动器通过远程访问服务器 245 从（服务器计算机 110 上的）软件镜像 210₁ 下载所选择的存储块。远程访问驱动器然后将所选择的存储块返回给部署代理 220（动作“A216a. 下载”）。部署代理 220 将所选择的存储块存储到当前镜像部分 225₁ 中（通过操作系统 255 的一个物理盘驱动器，图中未示出）；特别地，所选择的存储块被存储到具有等于其镜像地址加上当前镜像部分 225₁ 的偏移量（该偏移量，在这种情况下为零，从偏移量表 240 中抽取）的存储地址的存储位置中；而且，部署代理 220 相应地更新位置图，以便指示所选择的存储块的可用性—即，通过断言相应的标志（动作“A217. 存储”）。这时候，部署代理 220 将所选择的存储块返回给文件系统驱动器，文件系统驱动器进而将其返回给应用程序（动作“A218. 返回”）。

[0032] 相反地,如图 2C 中所示,如果所选择的存储块在当前镜像部分 225₁中已经可用(即位置图中其标志被断言),则部署代理 220 将该请求传给物理盘驱动器;物理盘驱动器直接从当前镜像部分 225₁,即从具有等于其镜像地址加上当前镜像部分 225₁的偏移量(在这种情况下为零)的存储地址的存储位置,检索所选择的存储块。然后物理盘驱动器将所选择的存储块返回给部署代理 220(动作“A216b. 检索”)。同样在这种情况下,部署代理 220 将所选择的存储块返回给文件系统驱动器,文件系统驱动器进而将其返回给应用程序(相同的动作“A218. 返回”)。

[0033] 上面描述的流式技术使得客户端计算机 115 在很短时间内准备好使用—就在新软件镜像 210₁的引导块已存储到大容量存储器 145c 中之后,甚至部署过程还在进行(例如,对于 10-200 兆字节典型大小的引导块在 1-2 分钟后);客户端计算机 115 的操作然后完全正常(与往常一样直接从大容量存储器 145c 启动),不管大容量存储器 145c 中软件镜像 210₁的其它存储块的可用性与否—当客户端计算机 115 访问还将要从服务器计算机 110 下载的存储块时,它的性能只有轻微下降。而且,使客户端计算机 115 准备好使用所需的时间不依赖于新软件镜像 210₁的大小。网络的使用还会随着时间而减少(例如,具有对数定律),因为一旦它们被访问一次,客户端计算机 115 上越来越多的存储块将变得可用。在这方面,应当说明的是为提供所要求的软件镜像,这个流式技术与本领域已知的那些技术无关。事实上,在已知的流式技术中,软件镜像的存储块只为了其立即的使用而被下载到客户端计算机上。然而,这些存储块不被永久存储在客户端计算机上(即,在它们被使用之后以及不管怎样在客户端计算机关机后会消失),因此为了它们的任何下一次使用,必须将它们再次下载;因此,客户端计算机从不与服务器计算机断开。事实上,即使当存储块被预先取出,它们在客户端计算机上也只保留到它们的(可能的)下一次使用;同样地,即使当存储块的本地缓存被实现,本地缓存中也只保留很少的存储块用于它们的再次使用(不管怎样,本地缓存中保留最少的最近使用过的存储块,它们最终被清除以便存储新的存储块)。

[0034] 现在参考图 2D,如果所选择的存储块已经被请求写入,部署代理 220 就将该请求传给物理盘驱动器;物理盘驱动器直接在当前镜像部分 225₁中,即,在具有等于其镜像地址加上当前镜像部分 225₁的偏移量(这种情况下为零)的存储地址的存储位置更新所选择的存储块,在该位置在其读取之后始终可用(动作“A219. 更新”)。因此,当前镜像部分 225₁能够被正常更新(就象新软件镜像 210₁已经全部部署到客户端计算机 115 上),甚至当部署过程还在进行中时。

[0035] 在完全异步方式中,部署代理 220 周期性地(例如,每 10-100 毫秒)检验服务器计算机 110、客户端计算机 115 和 / 或连接它们的网络的工作负载;如果该工作负载低于一个预定义的阈值(指示相应的资源那时未被充分利用—例如,因为服务器计算机 110 和 / 或客户端计算机 115 上没有执行动作,以及网络中的流量低),部署代理 220 通过重复上面描述的相同操作,下载在当前镜像部分 225₁中仍然不可用的一个新的存储块(例如,位置图中其标志被取消断言的第一个)(动作“A220. 下载”)。同上,部署代理 220 将这个新的存储块存储到当前镜像部分 225₁中(通过物理盘驱动器),即存储到具有等于其镜像地址加上当前镜像部分 225₁的偏移量(这种情况下为零)的存储地址的存储位置中;而且,部署代理 220 相应地更新位置图,以便指示新的存储块的可用性(动作“A221. 存储”)。

[0036] 在此方式中,如图 2E 所示,可保证新软件镜像 210₁的所有存储块最终都将被下载

(即使它们从未被使用)。

[0037] 在任何时候,操作员 212 还可以选择另一个新的软件镜像 210₁(例如,软件镜像 210₂),以通过部署管理器 205 将它们部署到同样的客户端计算机 115 上(动作“A205. 选择”)。作为响应,与上面一样,部署代理 220 通过远程访问服务器 245 从服务器计算机 110 下载新软件镜像 2012 的引导块(动作“A206. 下载”)。

[0038] 转到图 2F,部署代理 220 重定位初始镜像部分 225₁的被存储在新软件镜像 210₂的引导位置中的存储块(即在具有等于新软件镜像 210₂的引导块的镜像地址的存储地址的存储位置中);具体地,这些存储块被移动到服务部分 230 中(到其专用重定位部分),此处它们被称为是重定位的存储块 260₂(动作“A210. 重定位”)。部署代理 220 然后将新软件镜像 210₂的引导块存储到其引导位置中—此处用黑灰色显示它们并用标记 250₂表示。以这种方式,同样将引导块 250₂准确设置在大容量存储器 145c 中期望能在引导序列期间找到它们的地方;然而,这不会导致初始镜像部分 225₁中任何的信息丢失,因为被取代的相应存储块被保存在服务部分 230 中(动作“A211. 存储”)。

[0039] 部署过程然后如上述继续。具体地,部署代理 220 识别一个空闲的镜像部分 225₁用于接收新软件镜像 210₂(即,讨论示例中的镜像部分 225₂)。结果,部署代理 220 相应地更新镜像表 235(通过断言镜像部分 225₂的标志以指示其被使用)和偏移量索引 240(通过将它设置到镜像部分 225₂的偏移量,以指示它是当前的一个);而且,部署代理 220 将引导块 250₂的镜像地址保存到镜像表 235 内与当前镜像部分 225₂相关联的记录中。部署代理 220 还在当前镜像部分 225₂中创建对应的位置图(图中未示出)(相同动作“A212. 配置”)。部署代理 220 然后将新软件镜像 210₂的引导块存储到当前镜像部分 225₂中,即存储到大容量存储器 145c 的具有等于它们的镜像地址加上当前镜像部分 225₂的偏移量的存储地址的存储位置中—此处用灰色显示它们(相同动作“A213. 存储”)。这时候,部署代理 220 关闭然后开启客户端计算机 115。

[0040] 因此,如图 2G 中所示,客户端计算机 115 的引导装载程序现在从它的引导块 250₂启动。这样,与引导块 250₂相对应的新软件镜像 210₂的操作系统的一部分(用标记 255₂表示)和部署代理 220 被装载到工作存储器 130c 中(相同的动作“A214. 本地启动”)。在客户端计算机 115 操作期间,每一个访问软件镜像 210₂的所选择的存储块的请求再次由部署代理 220 的流式驱动器来服务。

[0041] 具体地,响应于访问(读取)新软件镜像 210₂的所选择的存储块的请求(相同动作“A215. 访问请求”),如果所选择的存储块在当前镜像部分 225₂中不可用,则从服务器计算机 110 上的软件镜像 210₂下载它(相同动作“A216a. 下载”),存储到当前镜像部分 225₂中(相同动作“A217. 存储”),然后返回(相同动作“A218. 返回”)。

[0042] 相反,如图 2H 所示,如果所选择的存储块在当前镜像部分 225₂中已经可用,则直接检索它(相同动作“A216b. 检索”),然后返回(相同动作“A218. 返回”)。

[0043] 现在参考图 2I,如果所选择的存储块已经被请求写入,则在当前镜像部分 225₂中更新它(相同动作“A219. 更新”)。当工作负载低时,下载当前镜像部分 225₂中还不可用的一个新的存储块(相同动作“A220. 下载”),并将其存储到当前镜像部分 225₂中(相同动作“A221. 存储”)。

[0044] 现在参考图 2J(显示了完全下载的软件镜像 210₂),在任何时候操作员 212 可以选

择另一个新的软件镜像 210_i (例如, 软件镜像 210_3), 以通过部署管理器 205 将其部署在同一个客户端计算机 115 上 (动作“A207. 选择”)。作为响应, 如上, 部署代理 220 通过远程访问服务器 245 从服务器计算机 110 下载新软件镜像 210_3 的引导块 (动作“A208. 下载”)。这时候, 部署代理 220 从服务部分 230 恢复被重定位的存储块 260_2 至初始镜像部分 225_1 —即进入当前镜像部分 225_2 的引导位置 (如镜像表 235 中对应的记录中所指示的)。这个操作不会导致被取代的当前镜像部分 225_2 的引导块 250_2 中的任何信息的丢失, 因为它们的最新的值 (如上面所解释, 在客户端计算机 115 的操作期间可能被更新) 被存储在当前镜像部分 225_2 中 (动作“A209. 恢复”)。

[0045] 然后部署过程如上述继续。具体地, 如图 2K 中所示, 部署代理 220 将初始镜像部分 225_1 的存储在新软件镜像 210_3 的引导位置中的存储块重定位至服务部分 230, 这里它们用标记 260_3 表示 (相同动作“A210. 重定位”)。这时候, 部署代理 220 将新软件镜像 210_3 的引导块存储到其引导位置中—这里它们用黑灰色显示并用标记 250_3 表示 (相同动作“A211. 存储”)。然后, 部署代理 220 识别一个空闲的镜像部分 225_i (即, 镜像部分 225_3) 来接收软件镜像 210_2 。接着, 部署代理 220 相应地更新镜像表 235 (通过断言镜像部分 225_3 的标志以指示其被使用) 和偏移量索引 240 (通过设置其为镜像部分 225_3 的偏移量以指示其是当前的一个); 而且, 部署代理 220 将引导块 250_3 的镜像地址保存到镜像表 235 中与当前镜像部分 225_3 相关联的记录中。部署代理 220 还在当前镜像部分 225_3 中创建对应的位置图 (图中未示出) (相同动作“A212. 配置”)。然后部署代理 220 将新软件镜像 210_3 的引导块存储到当前镜像部分 225_3 中—此处它们用灰色显示 (相同动作“A213. 存储”)。这时候, 部署代理 220 关闭然后开启客户端计算机 115, 以从其引导块 250_3 启动它, 由此装载与引导块 250_3 相对应的软件镜像 210_3 的操作系统的一部分和部署代理—它将处理每一个访问新软件镜像 210_3 的任何所选择的存储块的请求—到工作存储器 130c 中 (相同动作“A214. 本地启动”)。

[0046] 无论如何, 一旦完成部署一个或多个软件镜像至客户端计算机上, 它就可以独立工作, 而不再需要任何的服务器计算机。然而, 在部署过程期间所使用的相同流式技术也可以被用来创建客户端计算机的快照 (每个快照由其软件镜像在一特定的时间点的一致备份副本形成)。

[0047] 特别地, 图 3A–图 3E 显示了一个协作图, 其表示了可以用来实现根据本发明实施例的一个快照过程的主要软件组件的角色 (整体用标记 300 表示)。

[0048] 从图 3A 开始, 考虑例如客户端计算机 115 的情形, 其中两个软件镜像被存储在镜像部分 225_1 和 225_2 中; 镜像部分 225_2 是当前一个, 其引导块 250_2 也存储在初始镜像部分 225_1 中的其引导位置中 (初始镜像部分 225_1 的对应的被重定位的存储块 260_2 被保存在服务部分 230 中)。客户端计算机 115 的用户可以提交一个命令, 用于在对应的源镜像部分 225_i —例如, 存储源软件镜像 210_i 的源镜像部分 225_1 中创建源软件镜像的新的快照 (动作“A301. 创建”)。作为其响应, 部署代理 220 首先识别一个空闲的目标镜像部分 225_i (例如, 示例中的镜像部分 225_3) 来接收源软件镜像 210_i 。然后部署代理 220 将整个的源软件镜像 210_i 复制到目标镜像部分 225_3 中。更具体地, 对于源软件镜像 210_i 的每一个当前的存储块 (具有对应的当前镜像地址), 如果源镜像部分 225_i 是最初的一个 (如本例中), 则部署代理 220 检验当前存储块是否已经被重定位到服务部分 230 (即: 当前镜像地址等于当前软

件镜像 225₂的一个引导位置的镜像地址,如镜像表 235 的记录中所指示);如果是,则将对应的被重定位的存储块 260₂复制到目标镜像部分 225₃中,其位于具有等于当前镜像地址加上目标镜像部分 225₃的偏移量的存储地址的存储位置中。相反(即:当当前存储块未被重定位,或者当源镜像部分总不是最初的一个时),将当前存储块从源镜像部分 225₁复制到目标镜像部分 225₃—即从具有等于当前镜像地址加上源镜像部分 225₁的偏移量的存储地址的存储位置,复制到具有等于当前镜像地址加上目标镜像部分 225₃的偏移量的存储地址的存储位置(动作“A302. 复制”)。一旦整个源软件镜像 210₁已被复制到目标镜像部分 225₃中,部署代理 220 就相应地更新镜像表 235—通过断言目标镜像部分 225₃的标志以指示其被使用,并保存源软件镜像 210₁的引导块的相同镜像地址(动作“A303. 更新”)。因此,创建软件镜像快照的操作非常缓慢(其长度与要被复制的源软件镜像的大小成比例)。

[0049] 转到图 3B,客户端计算机 115 的用户还可以提交一个命令,用于删除一个过时的(obsolete)软件镜像(在对应的过时镜像部分 225₁中)至部署代理 220;为了确保在删除过时的软件镜像之后客户端计算机 115 的操作的连续性,过时的镜像部分 225₁应当不同于当前镜像部分 225₂—例如,镜像部分 225₁(动作“A304. 删除”)。作为其响应,部署代理 220 简单地更新镜像表 235—通过断言过时的镜像部分 225₁的标志以指示其是空闲的(动作“A304. 更新”)。因此,删除软件镜像的操作非常快速(因为它不需要其存储块上任何实际的动作);而且,这个操作不会导致任何问题,即使要被删除的镜像部分是最初的一个(因为存储在那里的当前镜像部分 225₂的引导块未受影响)。

[0050] 现在参考图 3C,客户端计算机 115 的用户可以替代地提交一个命令,用于从当前软件镜像 225₂切换至另一个(前一个)软件镜像 225₁—例如,软件镜像 225₃(动作“A305. 切换”)。作为其响应,如上,部署代理 220 将重定位的存储块 260₂从服务部分 230 恢复到初始镜像部分 225₁—即在当前镜像部分 225₂的引导位置(动作“A306. 恢复”)。

[0051] 转到图 3D,部署代理 220 将初始镜像部分 225₁的存储在前一个软件镜像 210₃的引导位置中的存储块重定位到服务部分 230,此处它们用标记 260₃表示(动作“A307. 重定位”)。然后部署代理 220 将前一个软件镜像 210₃的引导块复制到它们的引导位置中—此处它们用灰色显示并用标记 250₃表示(动作“A308. 复制”)。部署代理 220 现在可以通过将偏移量索引 240 设置为前一个镜像部分 225₃的偏移量以指示它现在是当前的一个来更新偏移量索引 240(动作“A309. 更新”)。这时候,部署代理 220 关闭然后开启客户端计算机 115,以便从其引导块 250₃启动它,由此将与引导块 250₃相对应的前一个软件镜像 210₃的操作系统的一部分和部署代理加载到工作存储器 130c 中(动作“A310. 重新启动(re-boot)”)。

[0052] 现在考虑图 3E,不管怎样,在客户端计算机 115 的操作期间,每一个访问当前软件镜像 210₃的所选择的存储块的请求都再次被部署代理 220 的流式驱动器服务(它取代操作系统 255₃的一个标准的文件系统驱动器,图中未示出)。

[0053] 具体地,文件系统驱动器接收一个访问(读或写)所选择的存储块的请求—例如,来自图中未示出的一个应用程序(动作“A311. 访问请求”)。该请求被传给部署代理 220,部署代理 220 在当前镜像部分 225₃中直接访问(通过操作系统 255₃的一个物理盘驱动,图中未示出)所选择的存储块,即,在具有等于所选择的存储块的镜像地址加上当前镜像部分 225₃的偏移量的存储地址的存储位置中(动作“A311. 访问”)。

[0054] 以上所描述的技术允许在客户端计算机上以非常容易的方式管理多个软件镜像。

而且,不同的软件镜像彼此间完全隔离;事实上,每个软件镜像只能访问对应的镜像部分(以便防止任何毁坏其它镜像部分的风险)。

[0055] 实现这个结果不需要任何虚拟化的设施;因此,客户端计算机的性能未受到不利的影响。事实上,在这种情形下,只有软件镜像(即存储它们的大容量存储器)被虚拟化;相反,操作系统在客户端计算机上继续自然地运行。

[0056] 特别地,这允许非常快速地恢复一个快照(因为它在客户端计算机上已经可用);而且,可以不需要任何网络连接来实现所需的结果。

[0057] 转到图 4,显示了一个表示可用来实现通用软件镜像的准备过程的主要软件组件的角色的协作图(整体用标记 400 表示),该通用软件镜像要在根据本发明的一实施例的方案中被使用。具体地,这个准备过程的目的在于识别软件镜像的引导块(要在以上描述的部署过程或快照过程期间被使用)。

[0058] 出于这个目的,服务器计算机 110 包括主软件镜像(或简称为主镜像)405 的贮藏库;每个主镜像 405 提供一个对应软件镜像的基础版本(例如,通过捕获其中先前安装了它的捐赠客户端计算机的硬盘的内容来创建),其中与捐赠客户端计算机的任何配置相关的特定内容(例如,驱动器和注册表设置)已经被移除。服务器计算机 115 还包括模型 410 的贮藏库;每个模型 410 替代地包括特定的内容,用于客户端计算机的对应配置。

[0059] 操作员 212 通过部署管理器 205 为一特定类型的客户端计算机(用相应的辅助客户端计算机 115 表示)选择一个软件镜像(包括一个所选择的主镜像 405 和一个所选择的模型 410)(动作“A401. 选择”)。作为其响应,部署管理器 205 通过传送所选择的软件镜像的标识符在辅助客户端计算机 115 上唤醒部署代理 220(动作“A402. 唤醒”)。接着,部署代理 220 安装所选择的软件镜像作为一个远程盘(例如,通过充当所讨论示例中的一个 iSCSI 启动器),用于通过远程访问服务器 245 来远程访问它。结果是,创建一个临时软件镜像(或简称为临时镜像)415,供辅助客户端计算机 145 专用访问;临时镜像 415 由指向所选择的主镜像 405 和所选择的模型 410 的存储块的索引结构来简单定义—即不做出它们的任何副本。临时镜像 415 与一个启用的块跟踪功能一起被安装,以便跟踪被访问的临时镜像 415 的任何存储块的镜像地址(动作“A403. 安装”)。

[0060] 这时候,部署代理 220 在临时镜像 415 上模拟辅助客户端计算机 115 的引导序列(直到加载部署代理);例如,在微软 Windows 中,部署代理 220 读取 MBR、引导扇区、bootmgr.exe 文件、boot\bcd 文件、系统注册表、winload.exe 文件、系统注册表中指定的驱动文件和部署代理(动作“A404. 模拟启动”)。一旦模拟引导序列已经完成,部署代理 220 就卸载临时镜像 415(动作“A405. 卸载”)。部署代理 220 然后将临时镜像 415 委托给部署管理器 205(动作“A406. 委托”)。作为响应,部署管理器 205 从临时镜像 415 建立一个新的软件镜像(或简称为新镜像)420(由其索引结构简单定义);而且,在模拟引导序列期间,新的镜像 420 与已被访问存储块的列表相关联,存储块将它定义为相应的引导块(动作“A407. 建立”)。

[0061] 当然,为了满足本地的具体的要求,本领域技术人员可将上述的方案应用到很多逻辑和/或物理修改和变更。更具体地,虽然参考其一个或多个实施例以一定程度的特殊性描述了这个方案,但是应当理解的是,形式或细节上的各种省略、替换和改变和其他实施例是可能的(例如,关于数值和组成)。特别地,为了提供更彻底的理解,甚至可以实行本

发明的不同实施例而不具有先前说明中所阐述的特定细节；相反地，为了不模糊那些具有不必要的细节的描述，省略或简化了已知的特征。而且，目的很明确，与所公开方案的任何实施例相关的特定元件和 / 或方法步骤可以包括在任何其它的实施例中，作为总体设计选择。

[0062] 例如，如果利用等同的方法（通过使用具有相同功能的类似步骤，它们是更多的步骤或它们的一部分，移除一些不重要的步骤或者增加更可选地步骤）实现相同方案，则类似考虑适用；而且，可以按不同的顺序、同时地或以交织方式（至少部分地）来执行这些步骤。

[0063] 软件镜像可以包括任何软件程序（例如，仅有操作系统而没有任何应用程序）；而且，即使所提出的技术被专门设计用于在物理的计算机上使用，同样不排除其在虚拟机上的应用。另外，可以使用任何等同的结构来提供访问功能。无论如何，也可以将同样的技术应用到具有任何引导序列的任何其它操作系统上。例如，在 Linux（是 LinuxTorvalds 的商标）中，引导块（除了部署代理外）将包括具有 GRBU 引导装载程序的 MBR，并且 /boot 目录包括内核和 initrd 文件系统；在这种情况下，在引导序列期间，BIOS 加载具有 GRBU 的 MBR，GRBU 查找 /boot 目录并加载内核和 initrd 文件系统，GRBU 在内核上启动，内核启动 initrd 文件系统，initrd 文件系统启动部署代理。

[0064] 当客户端计算机被切换至另一个软件镜像时恢复重定位的存储块的步骤不是绝对必要；例如，也能够不同于初始软件镜像的所有软件镜像的引导位置中重定位存储块，然后仅当客户端计算机被切换到初始软件镜像时同时恢复所有存储块（由此使得切换操作更快速，但是将浪费大容量存储器空间）。

[0065] 无论如何，引导位置可以被安排在大容量存储器内的任何位置。

[0066] 可以用等同的方式来管理镜像部分；例如，镜像部分可以具有不同的大小（甚至在创建它们时动态定义）。

[0067] 而且，可以使用等同的结构来管理镜像部分的状态信息（例如，对于直接包括其可用性的指示符的每一个镜像分区）。

[0068] 虽然在先前描述中，参考了当一个或多个软件镜像被完全安装时在客户端计算机上实现的一个快照过程（因此不需要网络连接），但是也可以将同样的技术应用于那些只部分地存储在客户端计算机的大容量存储器上的软件镜像（部署了当客户端计算机被反转到时重新启动的软件镜像）。

[0069] 在另一个实现中，部署代理通过使相应的操作系统相信其大小等于一个当前软件镜像来管理当前镜像部分（没有任何的其尺寸调整，但有相应的大容量存储器空间的浪费）。

[0070] 可以利用上面描述的流式技术从任何外部源（例如，可移动的存储设备）部署软件镜像；另外，可以使用任何等同的结构来管理大容量存储器中存储块的可用性（例如，通过将位置图分割成块，以允许将它们装载到客户端计算机的工作存储器中）。可选择地，还可以维持流式过程总是激活状态，甚至在软件镜像已经完全下载后（例如，为响应位置图中相应标志的复位，下载存储块的最新版本）。

[0071] 无论如何，所提出的技术完全无关于软件镜像如何被部署在客户端计算机上（例如，即使采用人工方式而不用任何服务器计算机）。

[0072] 不排除在部署相应的软件镜像期间将引导块只复制到它们的引导位置中的可能性（仅当客户端计算机被切换至另一个软件镜像时，才保存它们到其镜像部分中）。

[0073] 可选择地，可以通过物理盘驱动器来直接管理存储块的写入（没有任何通过部署代理的通路）。

[0074] 可以以任何其它的频率或只在特定时期内（例如，在晚上）监视工作负载；如果只针对客户端计算机、服务器计算机、网络或它们的任何组合来监视工作负载，则类似考虑适用。而且，可以以任何其它的方式来定义工作负载的阈值（例如，通过不同的权重来衡量它的贡献）。如果在下载两个或多个存储块的同时，工作负载下降到阈值之下，则类似考虑适用。无论如何，当流式过程总是激活时，可以省略这个特征。

[0075] 也能用不同方式来准备软件镜像（例如通过实际启动辅助客户端计算机，并跟踪在引导序列期间被访问的存储块，以识别其引导块）。

[0076] 如果用不同方式来组织该程序（可使用该程序来实现本发明的每个实施例），或提供附加的模块或功能，则类似考虑适用；同样地，存储器结构可以是其它类型，或者可以用等同的实体来替代（不必要包括物理存储介质）。该程序可以采用适合由任何数据处理系统使用或与其连接的任何形式（例如，在一虚拟机内）；特别地，该程序的形式可以是外部或驻留的软件、固件或微码（目标代码或源代码—例如，要被编译或翻译的）。而且，可以将该程序作为在任何计算机可使用的介质上实现的制造品来提供；该介质可以是任何适合包含、存储、通信、传播或传输程序的元件。例如该介质可以是电子的、磁性的、光学的、电磁的、红外的或半导体的类型；这样介质的例子有固定盘（可以在其中预加载程序）、可移动盘、磁带、卡片、电线、光纤、无线连接、网络、广播波等等。无论如何，可以利用一个硬件结构（例如，集成在半导体材料的芯片中）或者适当编程的或以其他方式配置的软件与硬件组合，来实现根据本发明实施例的方案。

[0077] 可选择地，该系统有不同的结构或者包括等同的组件，或者它具有其它可操作的特性。无论如何，可以将它们的每一个组件分割成多个元件，或者可以将两个或多个组件一起组合到单个元件中；而且，可以复制每个组件，来支持并行执行相应的操作。还要指出的是，不同组件之间的任何交互通常不需要连续（除非另外指定），可以是直接交互，也可以是通过一个或多个媒介间接交互。特别地，该系统可以是基于一个不同的结构（例如，广域、全球、蜂窝或卫星网络），并开发任何类型（有线和 / 或无线）的连接。无论如何，每个计算机可以有另一个结构或可以包括相似的元件（例如用来临时存储程序或其部分的高速缓冲存储器）；而且，可以利用任何代码执行实体（如 PDA、移动电话等等）或多个实体的组合来替代该计算机。

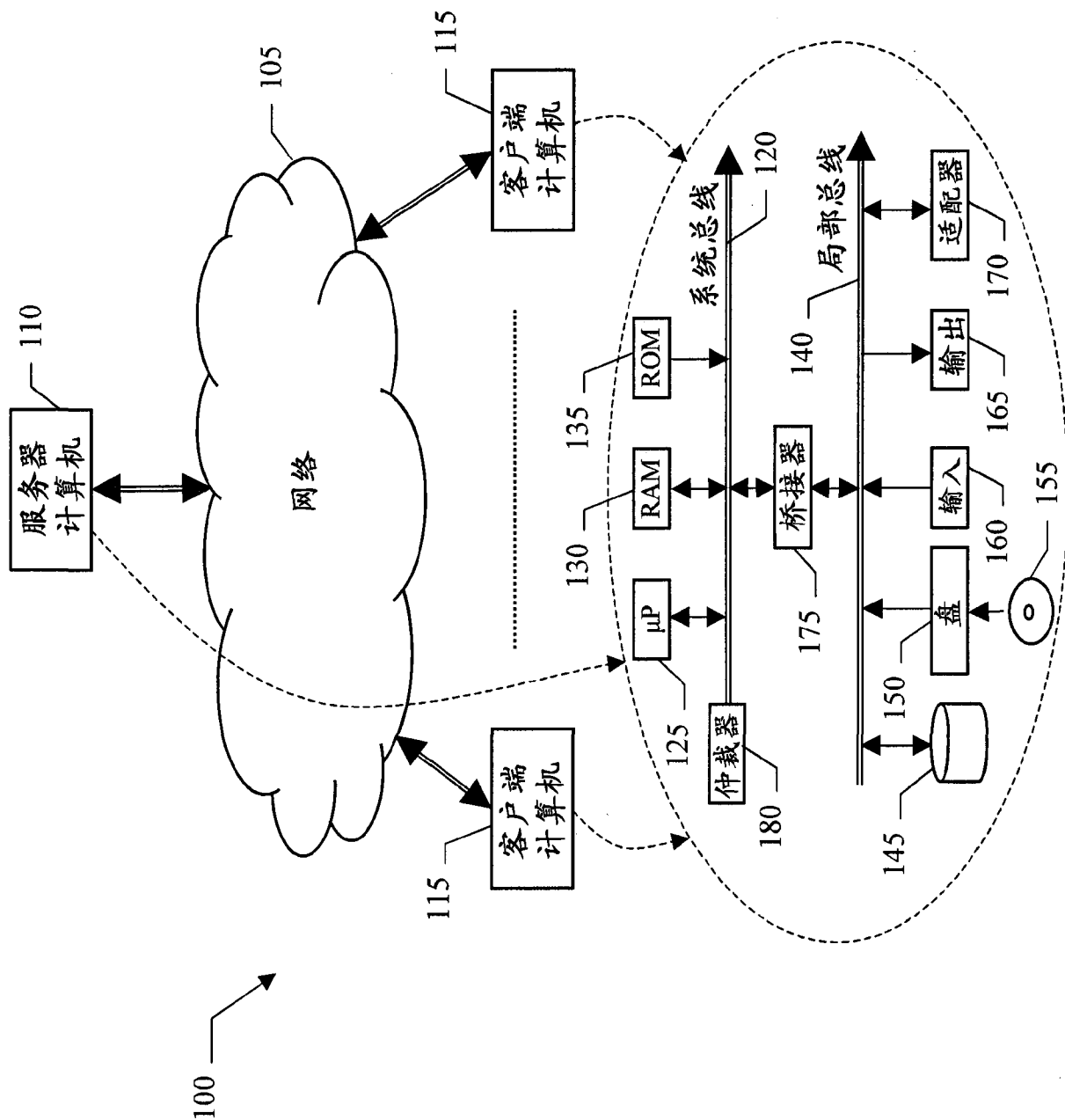


图 1

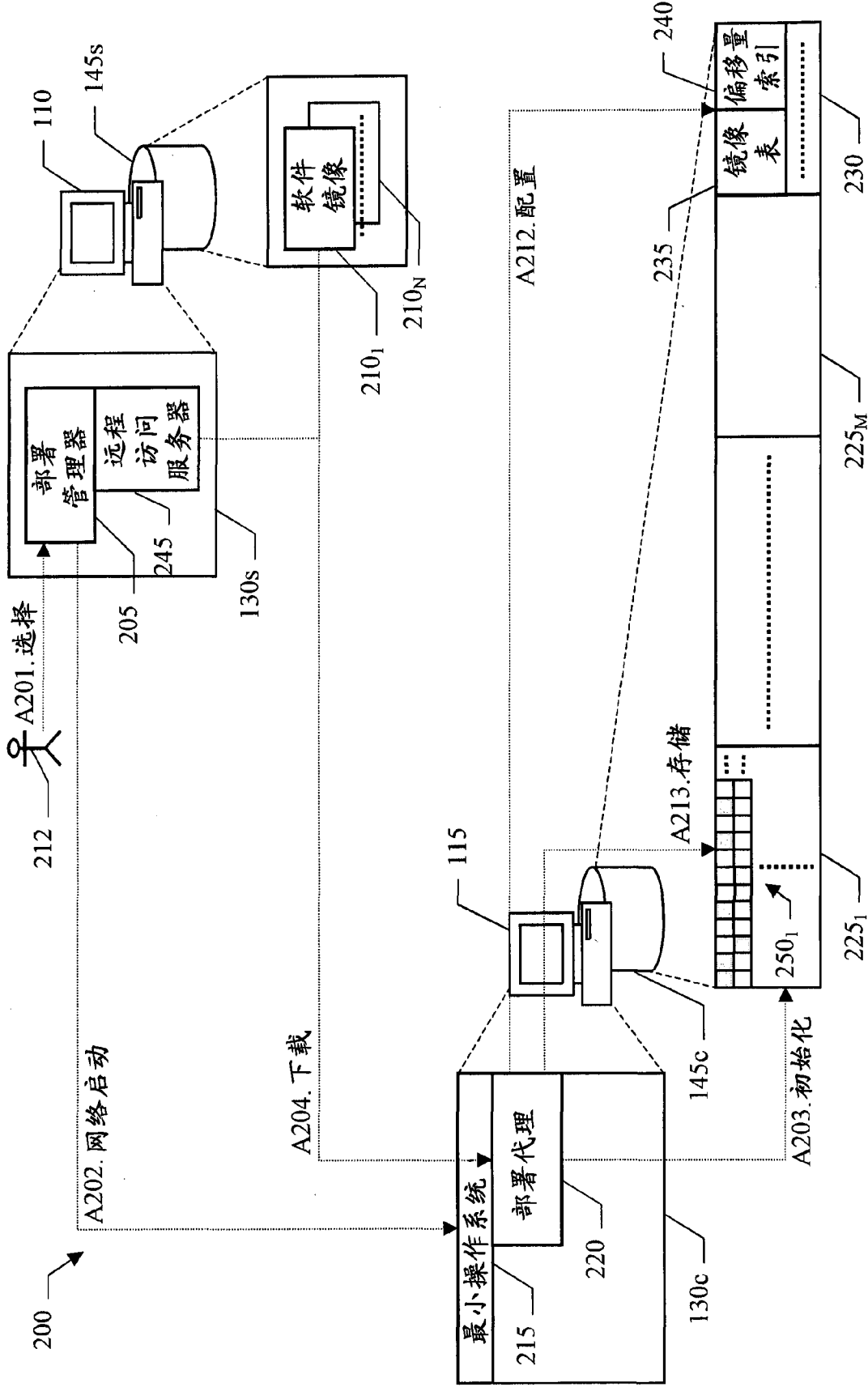


图 2A

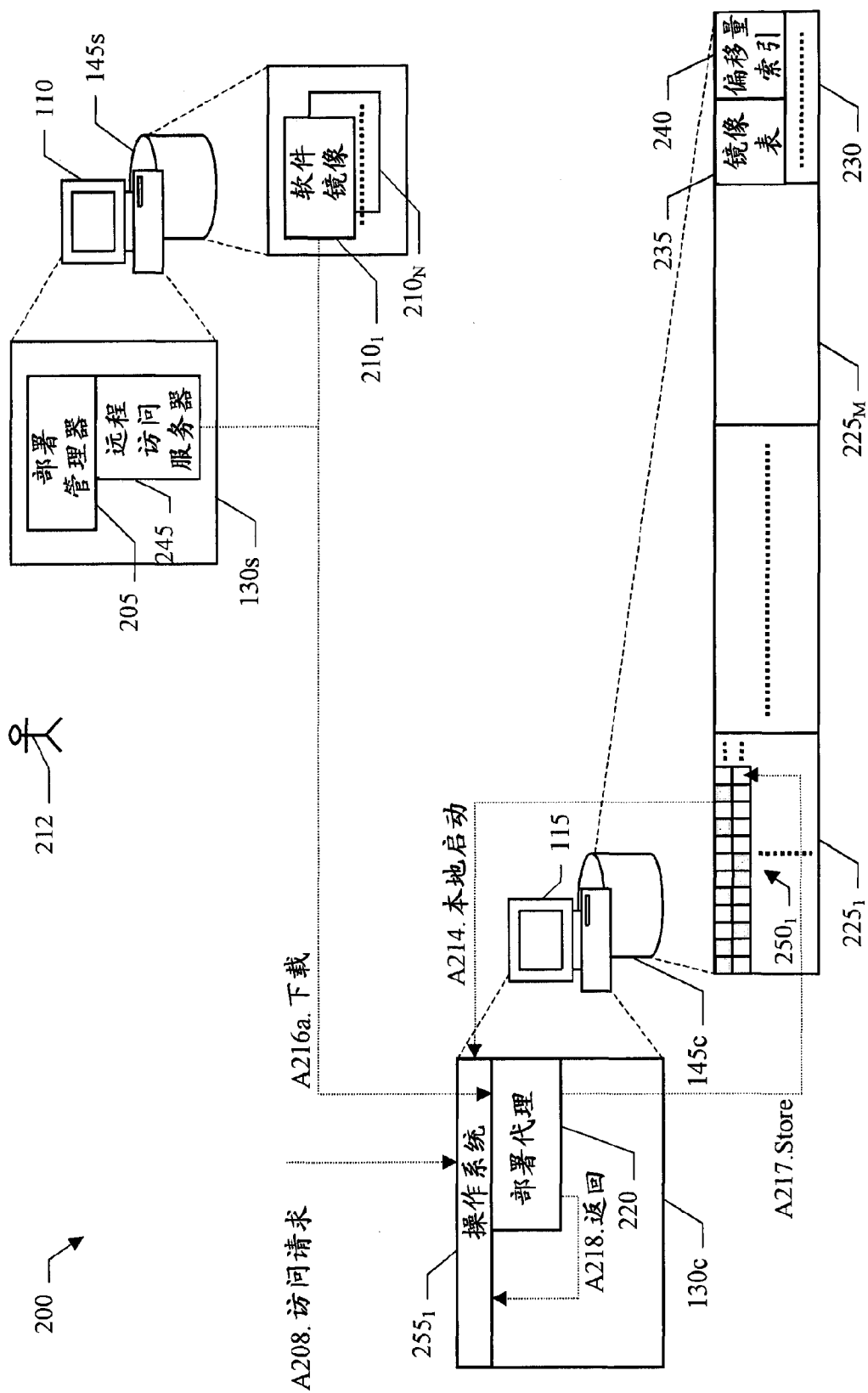


图 2B

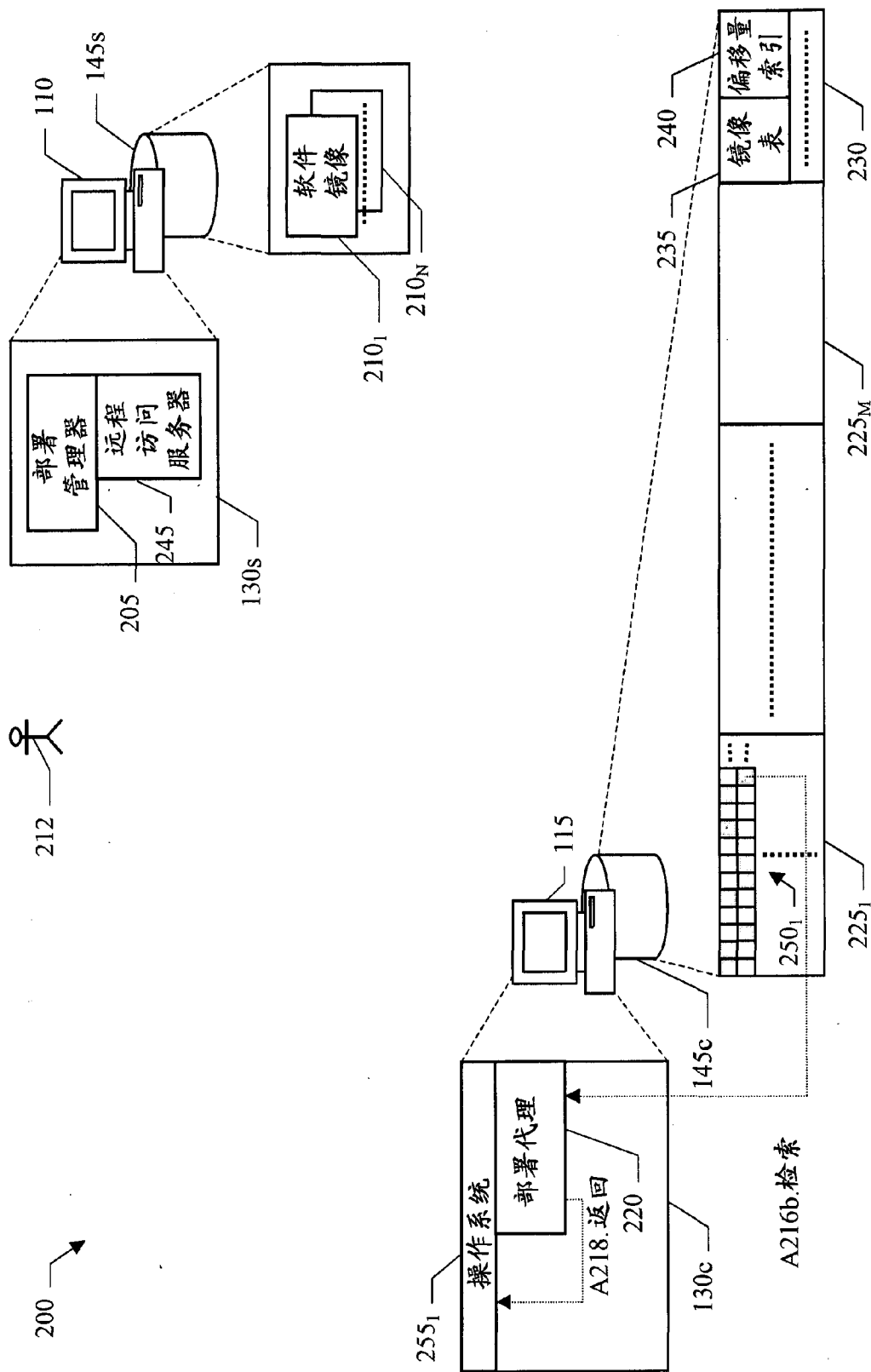


图 2C

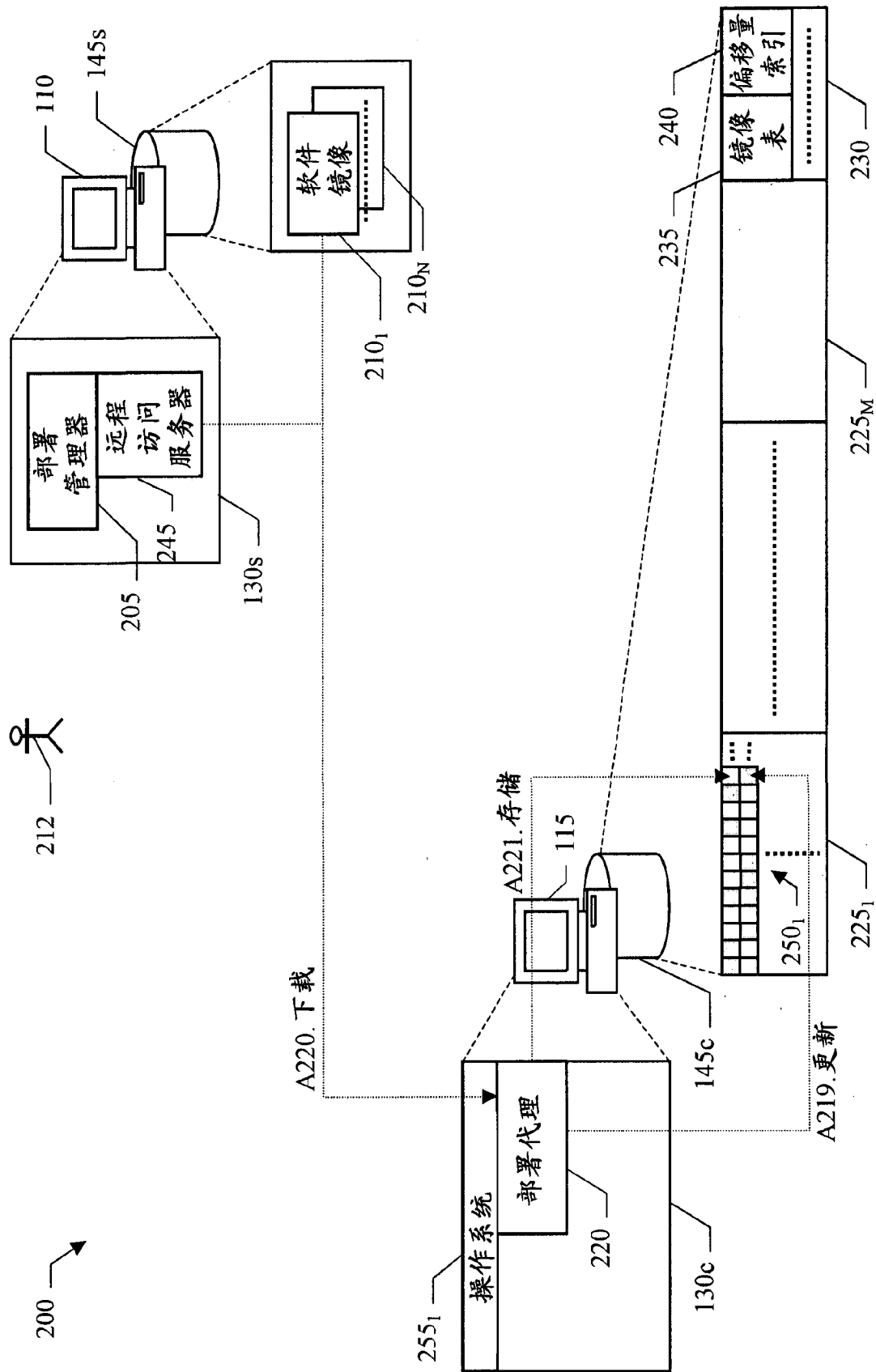


图 2D

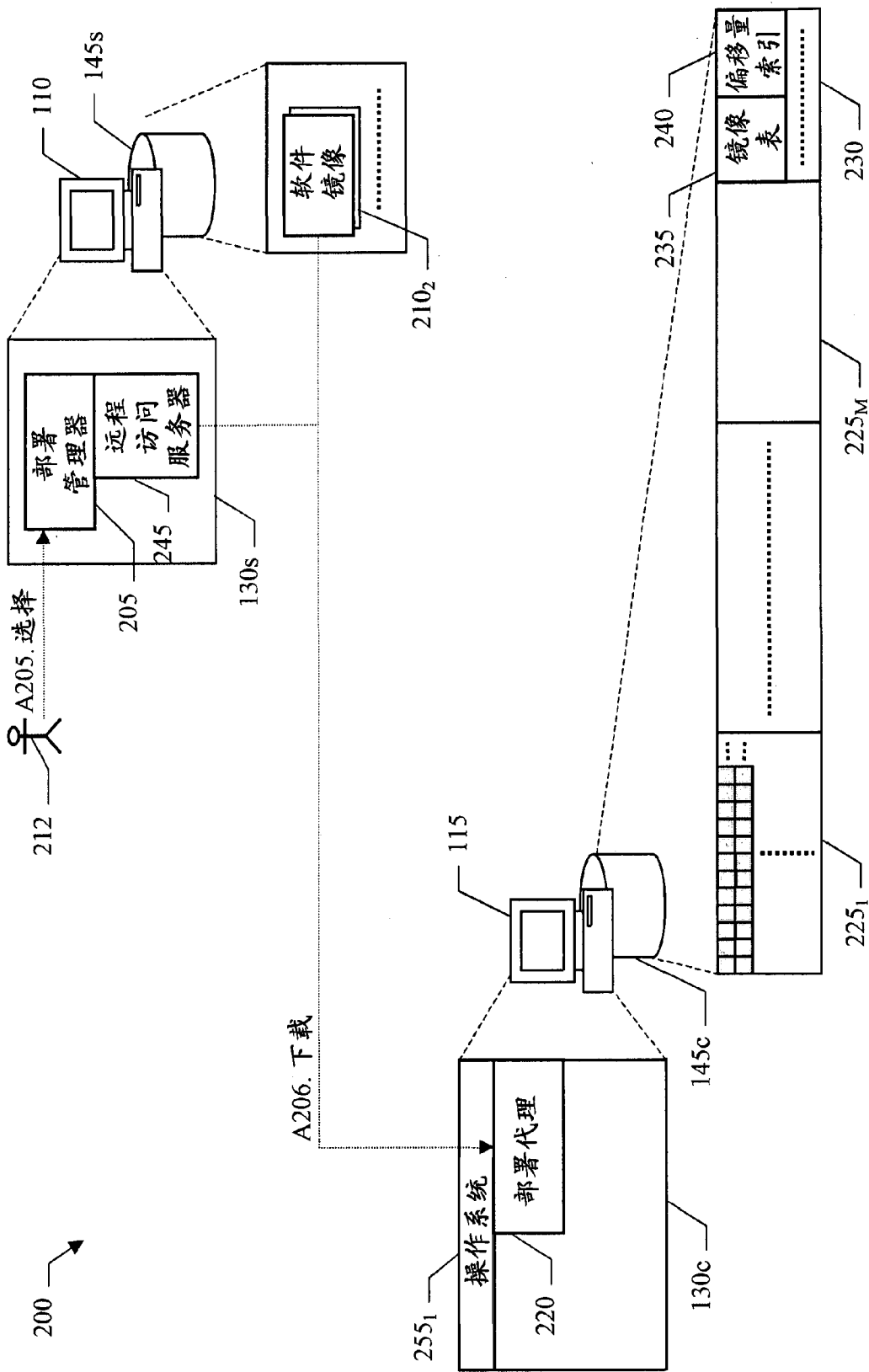


图 2E

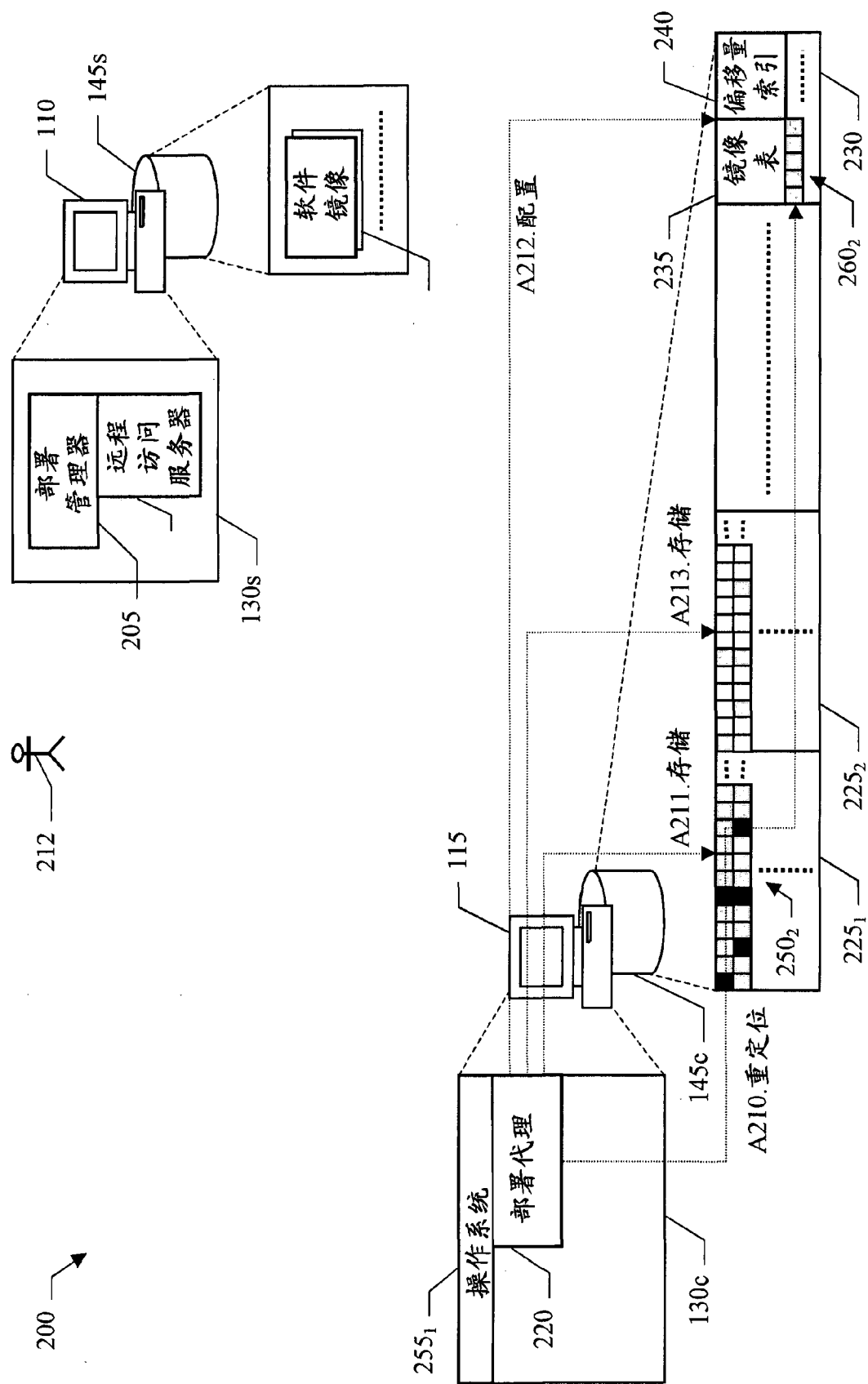


图 2F

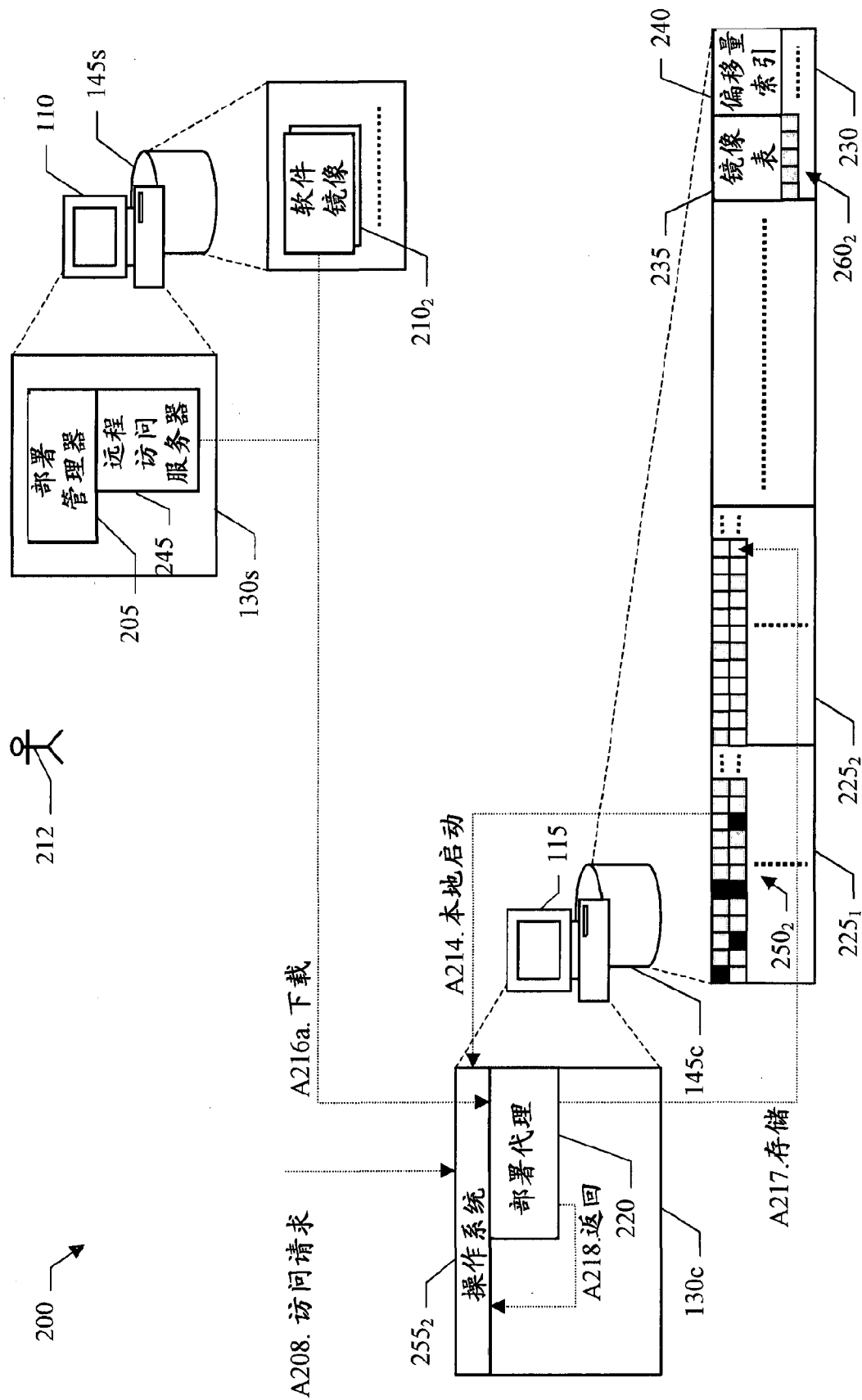


图 2G

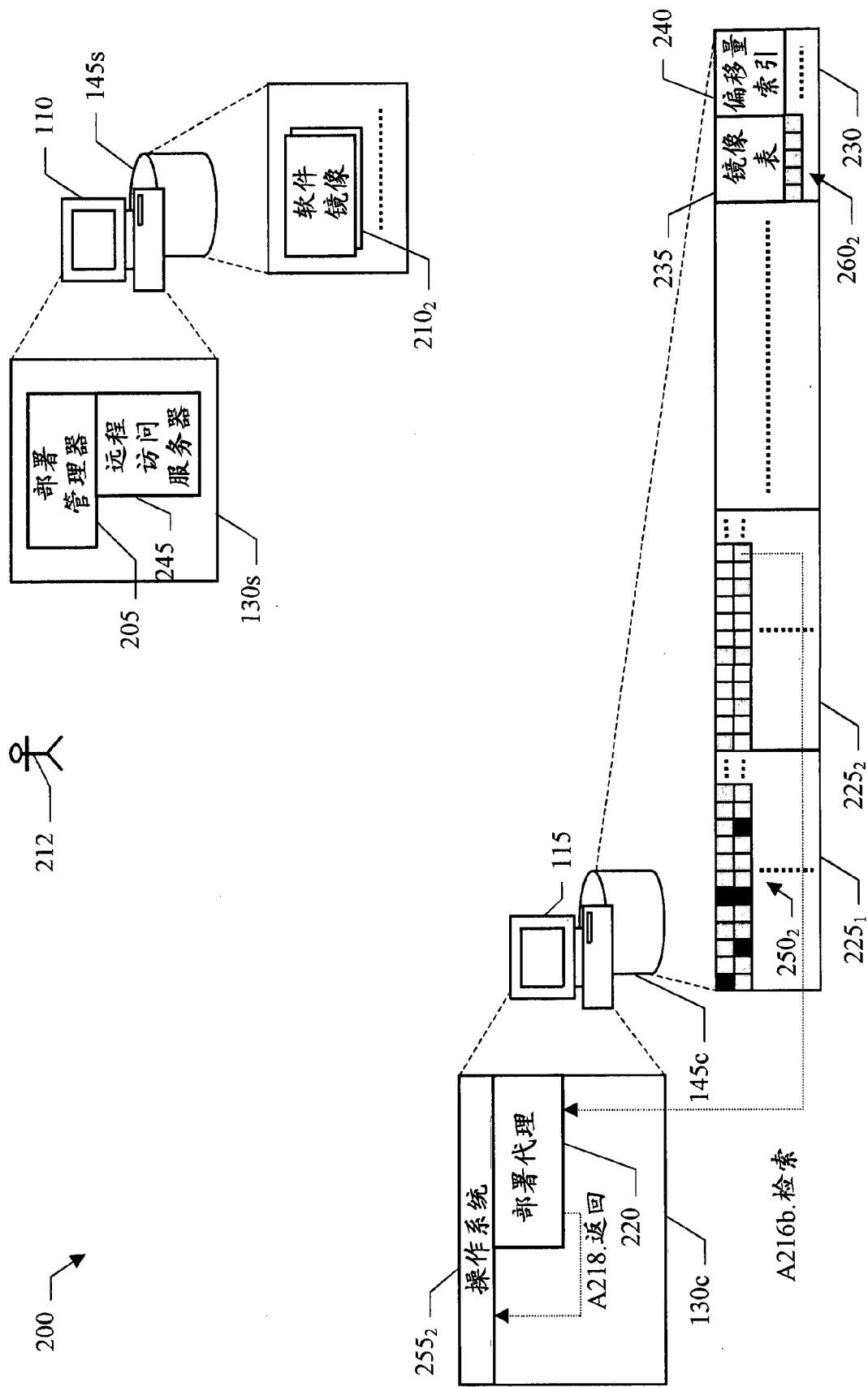


图 2H

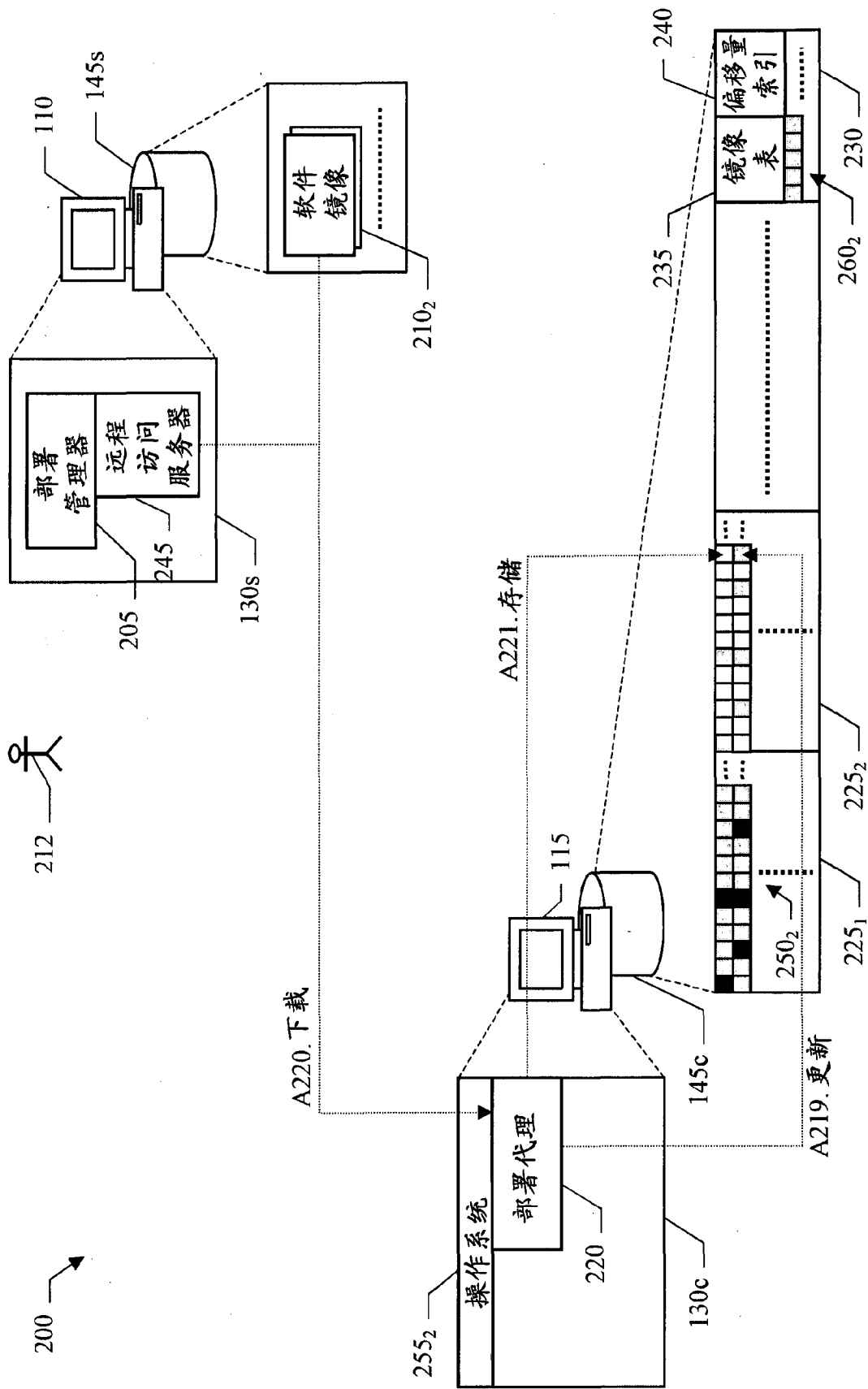


图 21

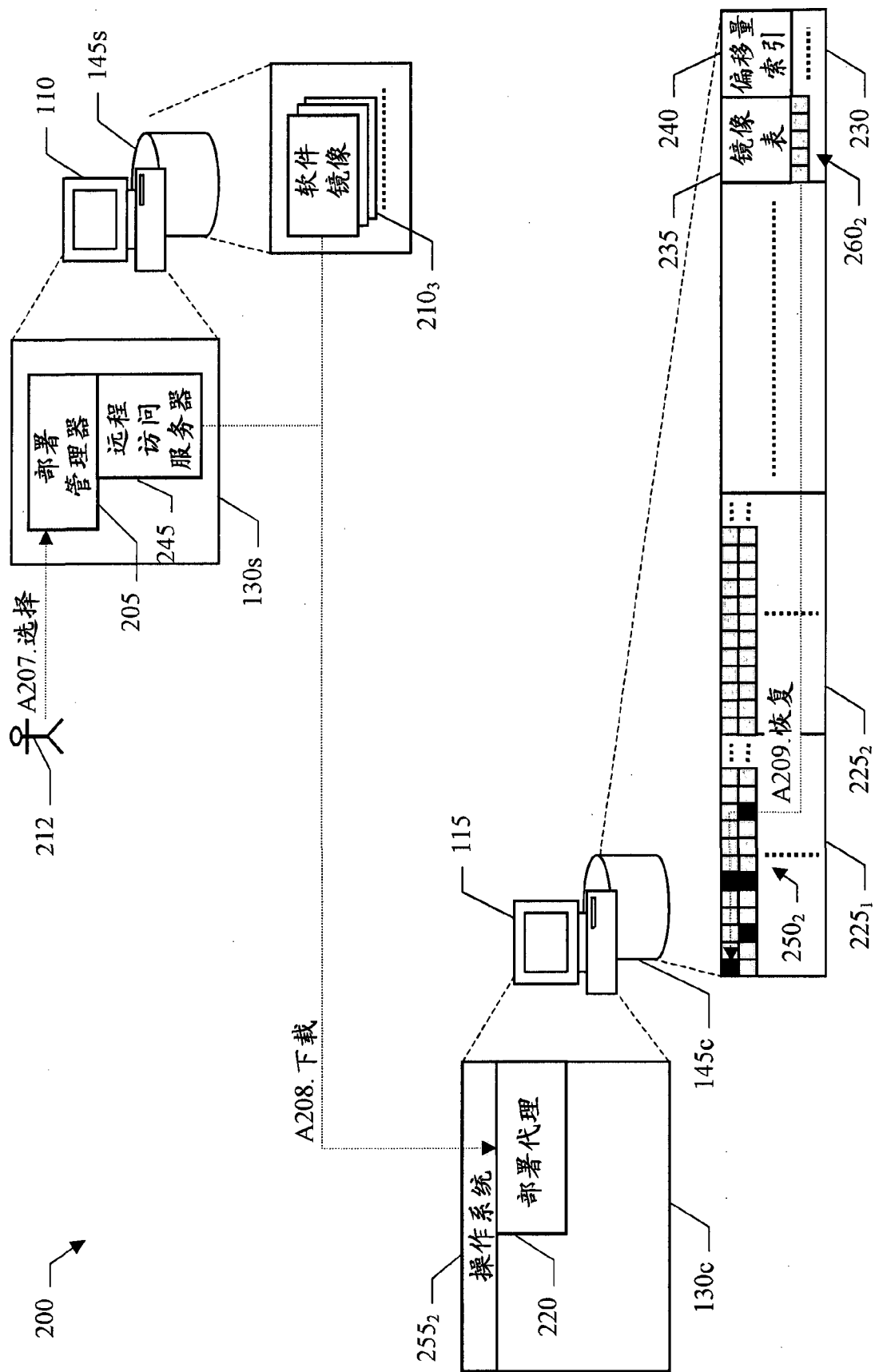


图 2J

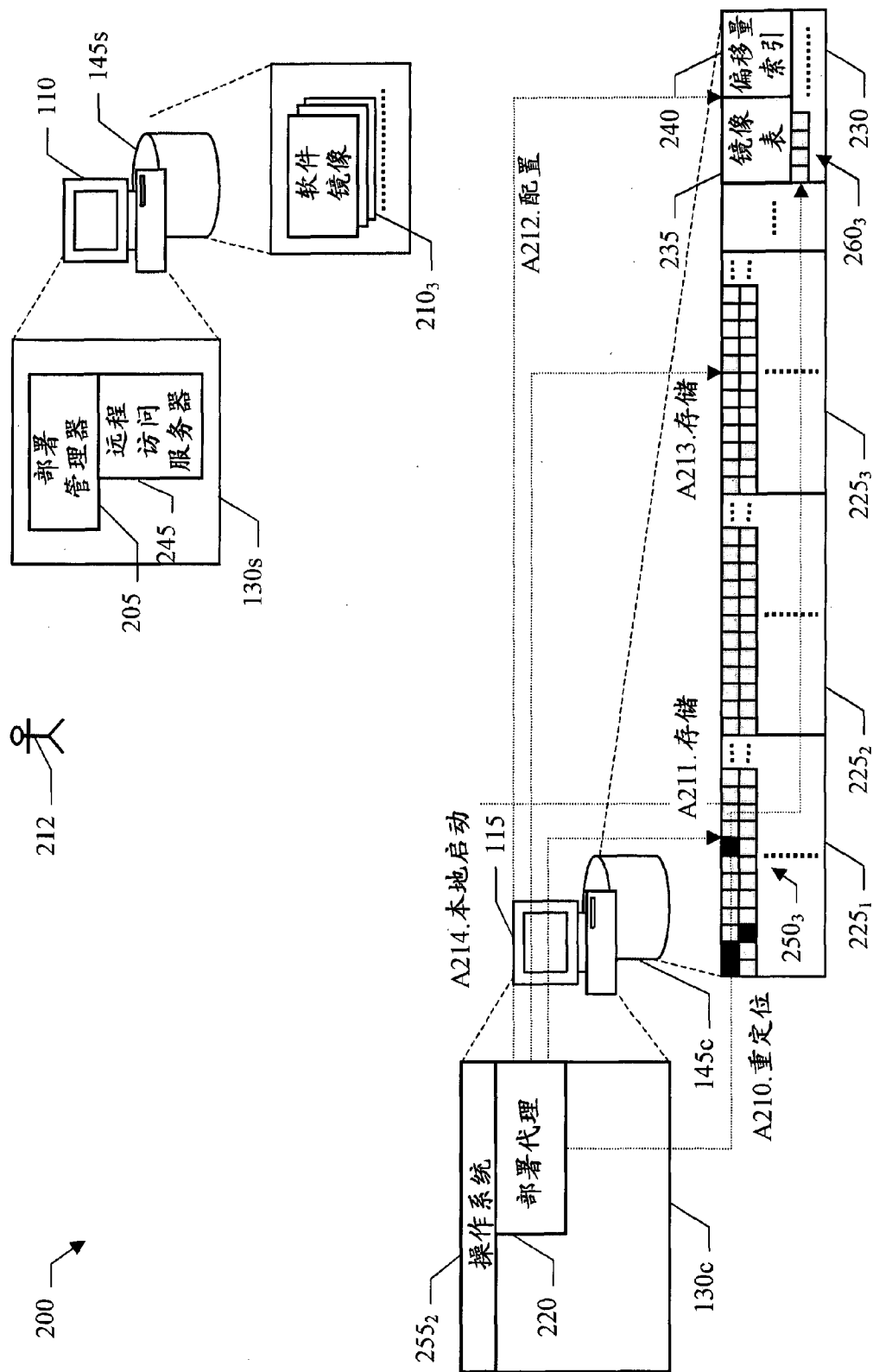


图 2K

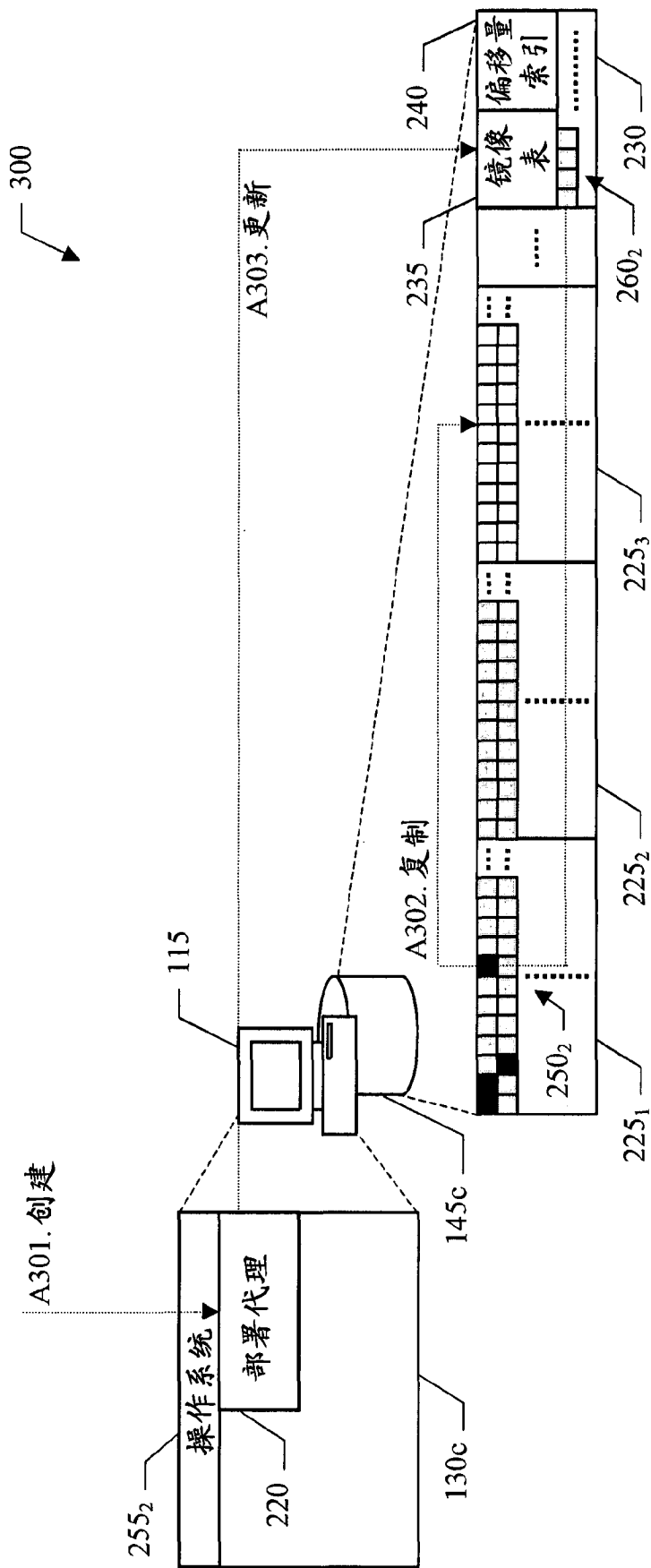


图 3A

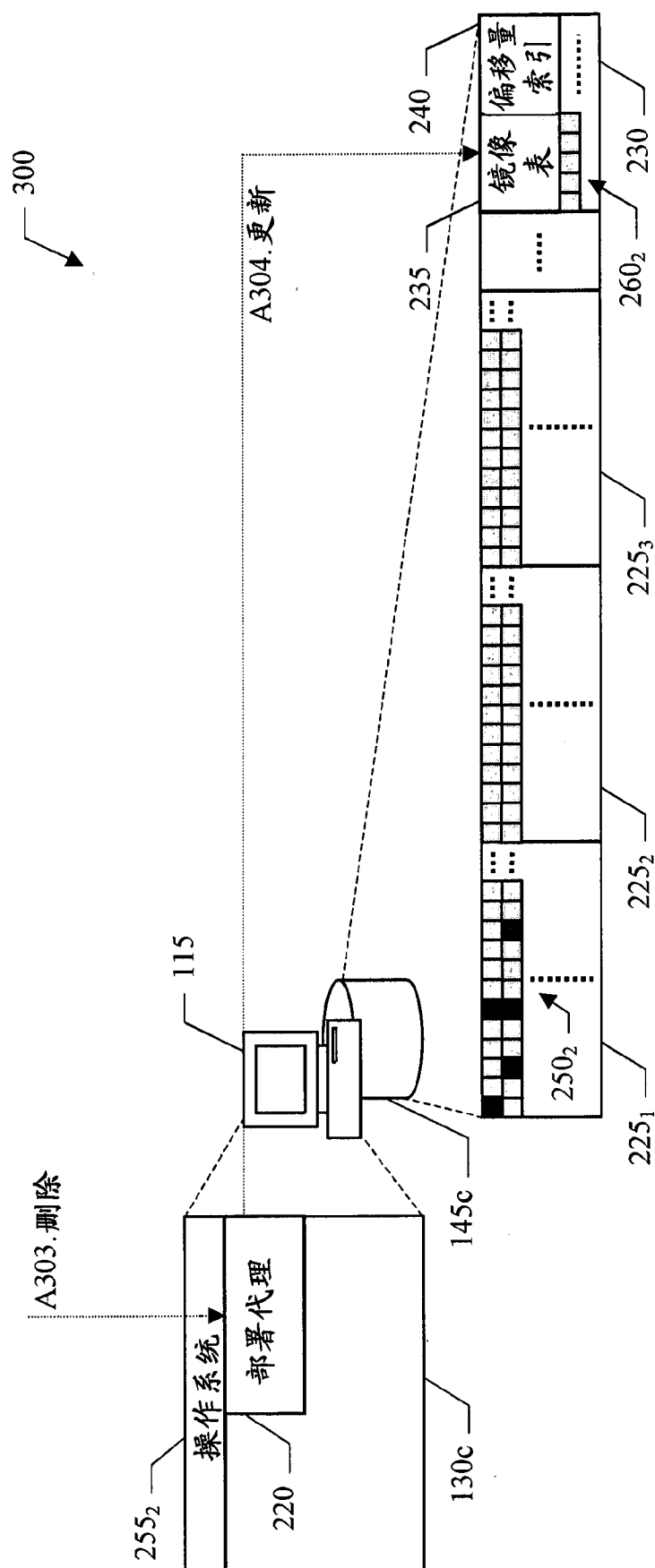


图 3B

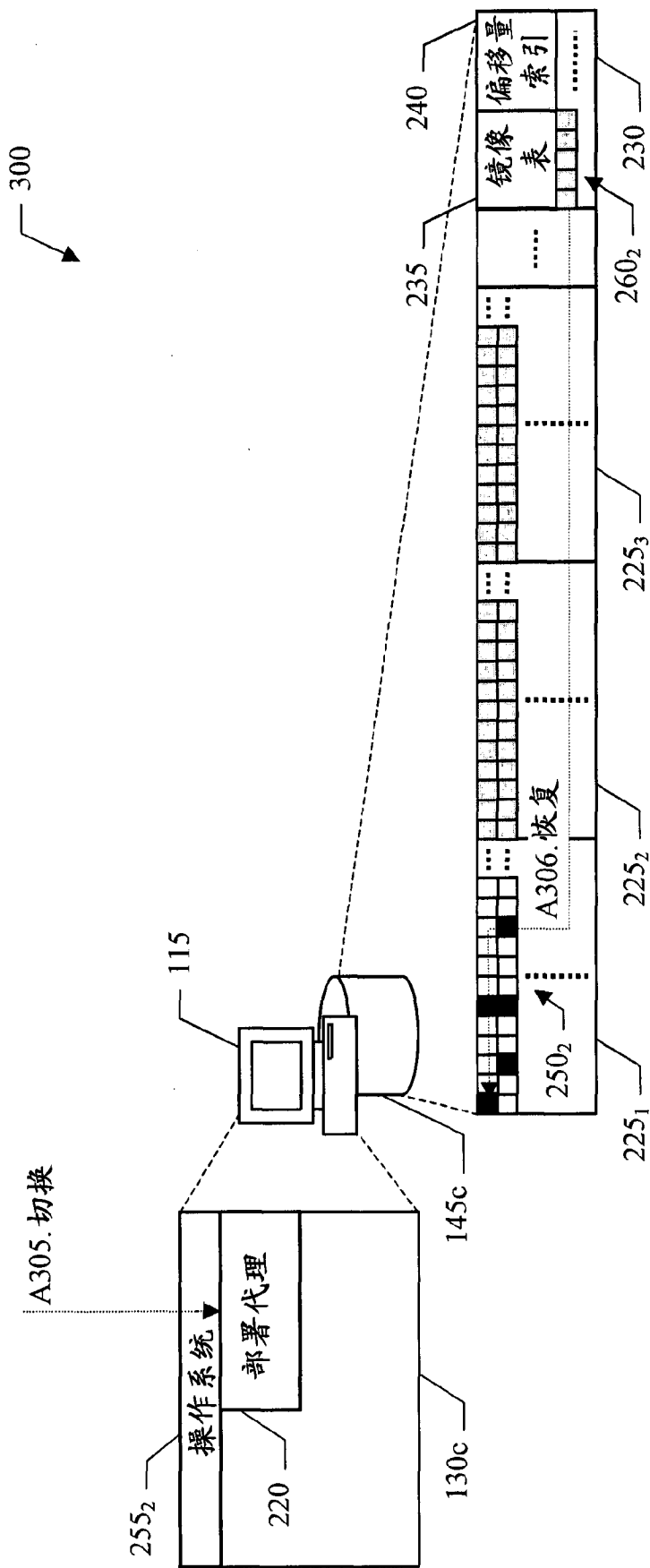


图 3C

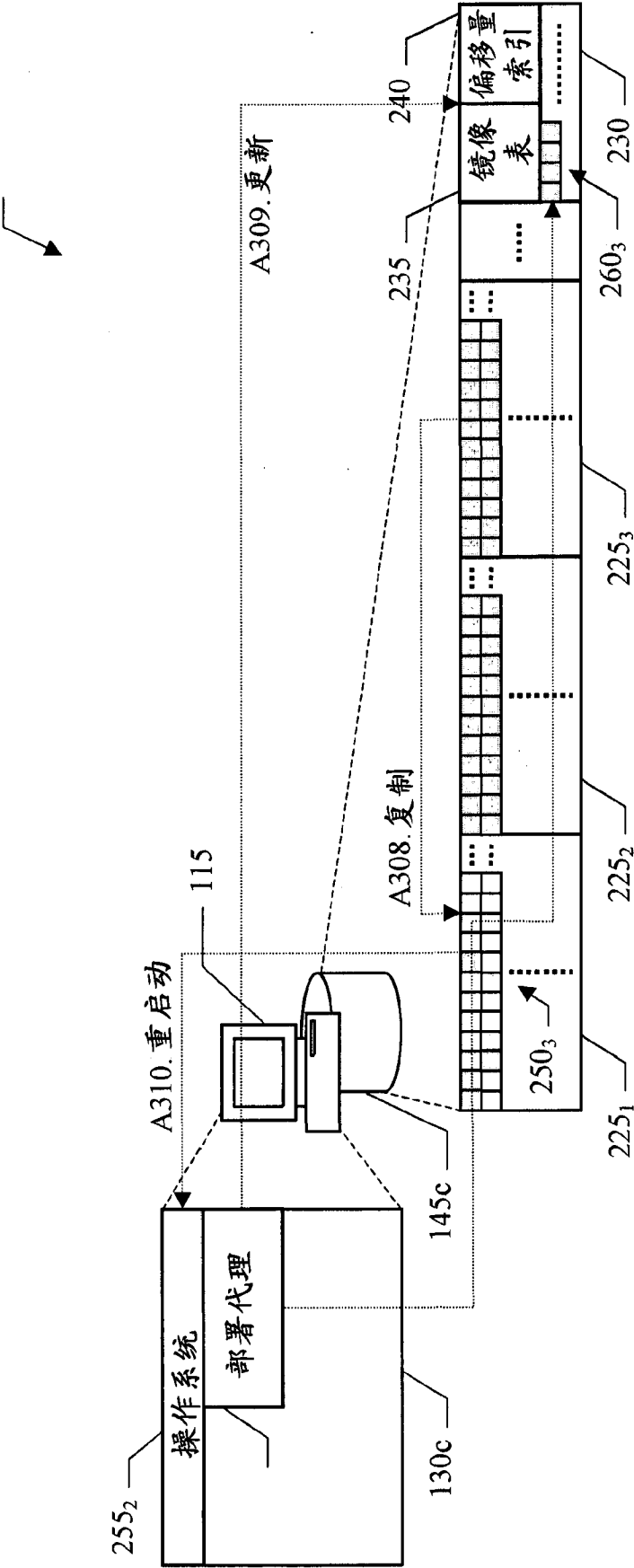


图 3D

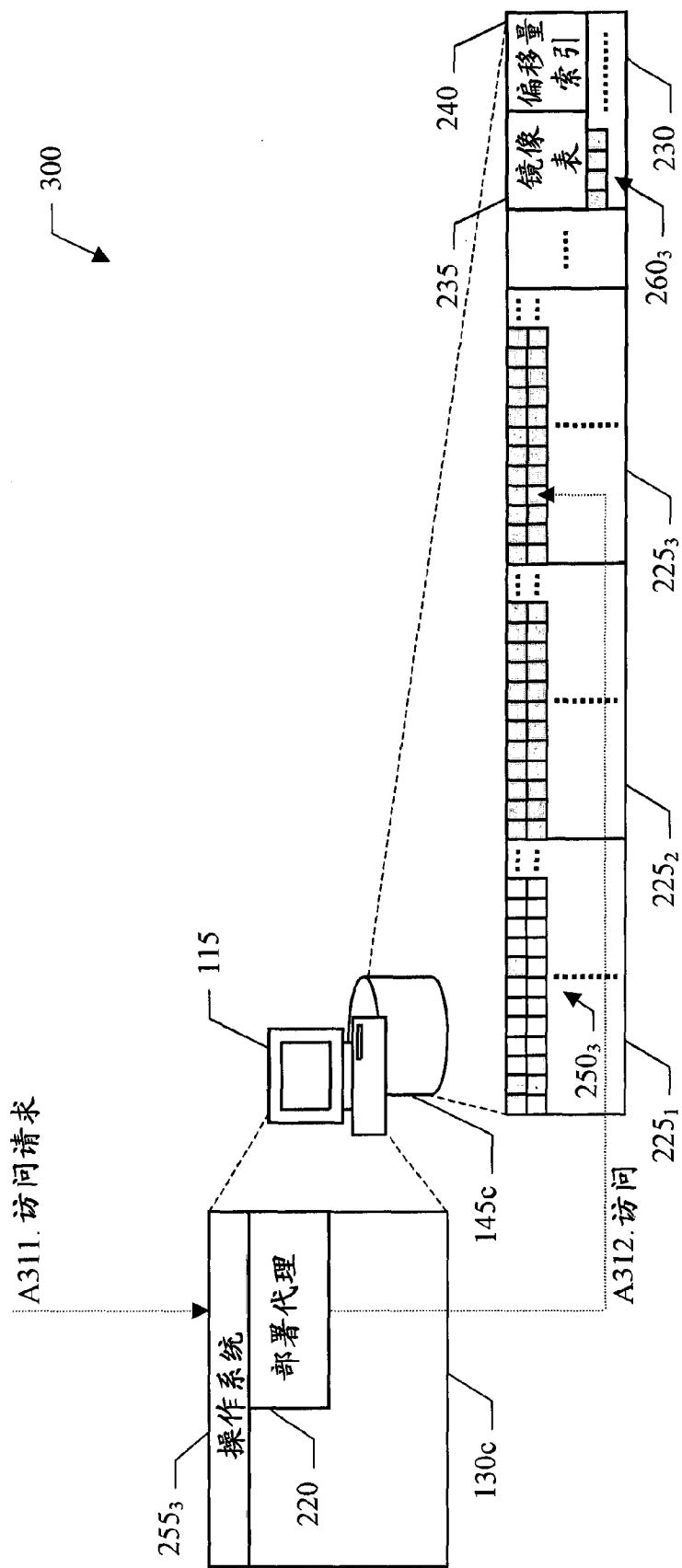


图 3E

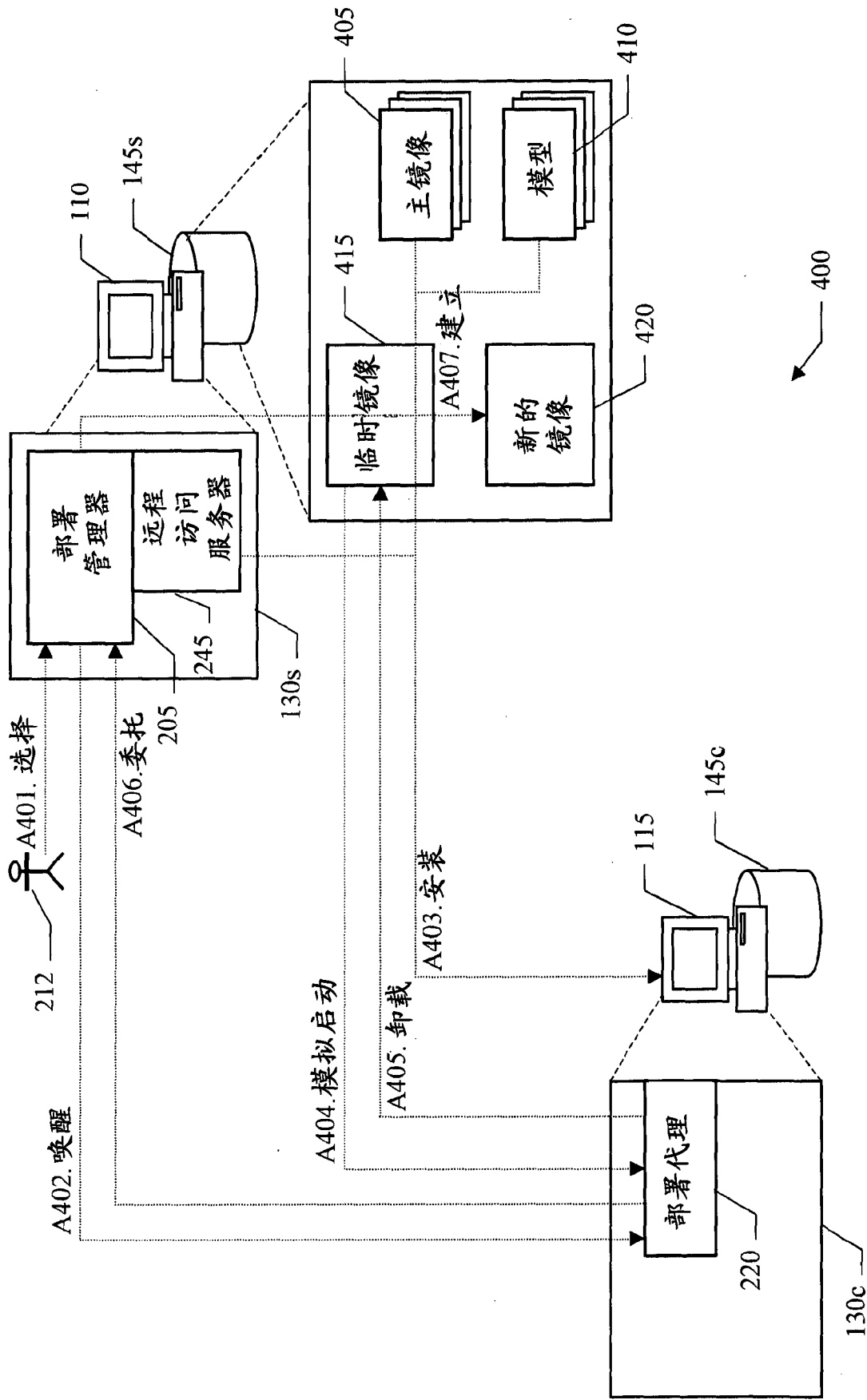


图 4