



# (12)发明专利

(10)授权公告号 CN 103782275 B

(45)授权公告日 2017. 10. 24

(21)申请号 201280042805.1

(22)申请日 2012.08.15

(65)同一申请的已公布的文献号  
申请公布号 CN 103782275 A

(43)申请公布日 2014.05.07

(30)优先权数据  
61/530,770 2011.09.02 US  
13/349,111 2012.01.12 US

(85)PCT国际申请进入国家阶段日  
2014.03.03

(86)PCT国际申请的申请数据  
PCT/US2012/050947 2012.08.15

(87)PCT国际申请的公布数据  
W02013/032711 EN 2013.03.07

(73)专利权人 高通股份有限公司

地址 美国加利福尼亚州

(72)发明人 诺曼·S·加尔加石  
维诺德·维贾雅拉詹

(74)专利代理机构 北京律盟知识产权代理有限  
责任公司 11287

代理人 宋献涛

(51)Int.Cl.  
G06F 9/445(2006.01)

(56)对比文件  
US 2002087734 A1,2002.07.04,  
US 2007150887 A1,2007.06.28,  
CN 102105848 A,2011.06.22,

审查员 张楚湖

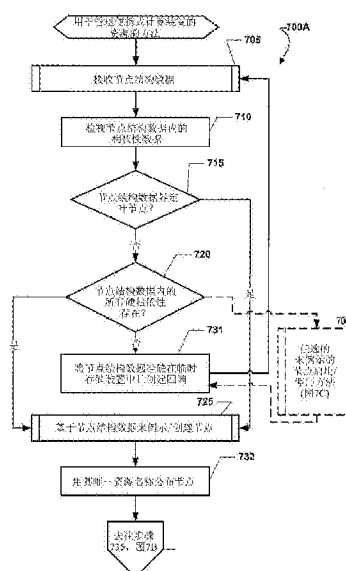
权利要求书4页 说明书23页 附图21页

## (54)发明名称

用于管理便携式计算装置的资源的方法和系统

## (57)摘要

本发明揭示一种用于管理便携式计算装置的资源的方法和系统。所述方法包含接收节点结构数据以用于形成节点,其中所述节点结构数据包含指派给所述节点的每一资源的唯一名称。一节点具有至少一个资源且其可具有多个资源。每一资源可为硬件或软件元件。所述系统包含框架管理器,所述框架管理器处置节点架构内的现有节点之间的通信。所述框架管理器还通过使用每一资源的唯一名称来记录每一资源的活动。所述框架管理器可将此所记录的活动发送到输出装置,例如打印机或显示屏。当将新硬件或软件元件(或两者)添加到便携式计算装置时,所述方法和系统可帮助减少或消除对自定义API的需要。



1. 一种用于管理具有由位于便携式计算装置中的多核心中央处理单元CPU的至少一个核心控制的多个装置资源的所述便携式计算装置的资源的方法,所述方法包括:

从所述便携式计算装置的硬件或软件元件接收节点结构数据以用于未例示的节点,所述未例示的节点表示或模拟所述硬件或软件元件,所述节点结构数据包括包含在所述便携式计算装置内的作为所述节点的部分的所述多个装置资源中的每一资源的唯一名称;

针对包含在所述便携式计算装置内的所述多个装置资源之间的一个或多个相依性来检视所述节点结构数据;

确定与所述未例示的节点的一个或多个相依性相关联的所述多个装置资源中的每一资源是否存在于用于所述便携式计算装置的节点框架内,所述节点框架由图表界定,其中所述图表的每一节点表示由所述便携式计算装置的所述CPU控制的一个或多个装置资源;

确定与所述未例示的节点相关联的所述多个装置资源中的每一资源是否可用于支持来自所述便携式计算装置内部的客户端请求,其中所述客户端包括所述便携式计算装置的硬件或软件元件;

如果用于所述一个或多个相依性的所述多个装置资源中的每一资源存在且可用,那么在所述多核心CPU的一个核心上例示所述未例示的节点;

创建针对与新例示的节点相关联的所述多个装置资源中的一个资源的阈值事件,其中创建所述阈值事件包括从所述多个装置资源中的其他资源或执行线程接收一个或多个请求以创建所述阈值事件,接收与所述阈值事件相关联的多个条件,接收跟踪与所述阈值事件相关联的所述多个条件的至少一个功能,以及接收所述多个装置资源中的每一资源的所述唯一名称,所述多个装置资源将接收发生所述阈值事件的通知;

确定与所述未例示的节点相关联的所述多个装置资源中的每一资源是否可用于支持各自的请求;以及

如果与所述未例示的节点相关联的所述多个装置资源中的任何资源不能用于支持各自的请求,则组织其他装置资源使用所述未例示的节点。

2. 根据权利要求1所述的方法,其中每一资源包括第二软件和第二硬件元件中的至少一者。

3. 根据权利要求1所述的方法,其进一步包括从所述便携式计算装置的硬件或软件元件接收数据以用于形成充当所述节点框架中的至少一个功能资源的构造的占位符的存根资源。

4. 根据权利要求3所述的方法,其进一步包括接收数据以用所述功能资源来取代所述存根资源。

5. 根据权利要求1所述的方法,其进一步包括确定所述客户端请求是否可压制以延迟执行。

6. 根据权利要求5所述的方法,其进一步包括确定是否已实现用于延迟所述可压制请求的执行的所述多个条件中的一个或多个。

7. 根据权利要求1所述的方法,其中所述阈值事件包括所述便携式计算装置的温度阈值。

8. 根据权利要求7所述的方法,其中与所述阈值事件相关联的所述多个条件包括所述便携式计算装置的多个温度。

9. 根据权利要求1所述的方法, 其中所述便携式计算装置包括移动电话、个人数字助理、寻呼机、智能电话、导航装置以及具有无线连接或链路的手持式计算机中的至少一者。

10. 一种用于管理具有由至少一个核心控制的多个装置资源的便携式计算装置的资源的计算机系统, 所述系统包括:

位于所述便携式计算装置中的多核心中央处理单元CPU, 所述多核心CPU的至少一个核心可操作以用于:

从所述便携式计算装置的硬件或软件元件接收节点结构数据以用于未例示的节点, 所述未例示的节点表示或模拟所述硬件或软件元件, 所述节点结构数据包括包含在所述便携式计算装置内的作为所述节点的部分的所述多个装置资源中的每一资源的唯一名称;

针对包含在所述便携式计算装置内的所述多个装置资源之间的一个或多个相依性来检视所述节点结构数据;

确定与所述未例示的节点的一个或多个相依性相关联的所述多个装置资源中的每一资源是否存在于用于所述便携式计算装置的节点框架内, 所述节点框架由图表界定, 其中所述图表的每一节点表示由所述便携式计算装置的所述CPU控制的一个或多个装置资源;

确定与所述未例示的节点相关联的所述多个装置资源中的每一资源是否可用于支持来自所述便携式计算装置内部的客户端请求, 其中所述客户端包括所述便携式计算装置的硬件或软件元件;

如果用于所述一个或多个相依性的所述多个装置资源中的每一资源存在且可用, 那么在所述多核心CPU的一个核心上例示所述未例示的节点;

创建针对与新例示的节点相关联的所述多个装置资源中的一个资源的阈值事件, 其中创建所述阈值事件包括从所述多个装置资源中的其他资源或执行线程接收一个或多个请求以创建所述阈值事件, 接收与所述阈值事件相关联的多个条件, 接收跟踪与所述阈值事件相关联的所述多个条件的至少一个功能, 以及接收所述多个装置资源中的每一资源的所述唯一名称, 所述多个装置资源将接收发生所述阈值事件的通知;

确定与所述未例示的节点相关联的所述多个装置资源中的每一资源是否可用于支持各自的请求; 以及

如果与所述未例示的节点相关联的所述多个装置资源中的任何资源不能用于支持各自的请求, 则组织其他装置资源使用所述未例示的节点。

11. 根据权利要求10所述的系统, 其中每一资源包括第二软件和第二硬件元件中的至少一者。

12. 根据权利要求10所述的系统, 其中所述多核心CPU的至少一个核心进一步可操作以用于:

接收数据以用于形成充当所述节点框架中的至少一个功能资源的构造的占位符的存根资源。

13. 根据权利要求12所述的系统, 其中所述多核心CPU的至少一个核心进一步可操作以用于:

接收数据以用功能资源来取代所述存根资源。

14. 根据权利要求10所述的系统, 其中所述多核心CPU的至少一个核心进一步可操作以用于:

确定所述客户端请求是否可压制以延迟执行。

15. 根据权利要求14所述的系统,其中所述多核心CPU的至少一个核心进一步可操作以用于:

确定是否已实现用于延迟可压制请求的执行的一个或一个以上条件。

16. 根据权利要求10所述的系统,其中所述阈值事件包括所述便携式计算装置的温度阈值。

17. 根据权利要求16所述的系统,其中与所述阈值事件相关联的所述多个条件包括所述便携式计算装置的多个温度。

18. 根据权利要求10所述的系统,其中所述便携式计算装置包括移动电话、个人数字助理、寻呼机、智能电话、导航装置以及具有无线连接或链路的手持式计算机中的至少一者。

19. 一种用于管理具有由位于便携式计算装置中的多核心中央处理单元CPU的至少一个核心控制的多个装置资源的所述便携式计算装置的资源的计算机系统,所述系统包括:

用于从所述便携式计算装置的硬件或软件元件接收节点结构数据以用于未例示的节点的装置,所述未例示的节点表示或模拟所述硬件或软件元件,所述节点结构数据包括包含在所述便携式计算装置内的作为所述节点的部分的所述多个装置资源中的每一资源的唯一名称;

用于针对包含在所述便携式计算装置内的所述多个装置资源之间的一个或多个相依性来检视所述节点结构数据的装置;

用于确定与所述未例示的节点的一个或多个相依性相关联的所述多个装置资源中的每一资源是否存在于用于所述便携式计算装置的节点框架内的装置,所述节点框架由图表界定,其中所述图表的每一节点表示由所述便携式计算装置的所述CPU控制的一个或多个装置资源;

用于确定与所述未例示的节点相关联的所述多个装置资源中的每一资源是否可用于支持来自所述便携式计算装置内部的客户端请求的装置,其中所述客户端包括所述便携式计算装置的硬件或软件元件;

用于在用于所述一个或多个相依性的所述多个装置资源中的每一资源存在且可用的情况下,在所述多核心CPU的一个核心上例示所述未例示的节点的装置;

用于创建针对与新例示的节点相关联的所述多个装置资源中的一个资源的阈值事件的装置,其中创建所述阈值事件包括从所述多个装置资源中的其他资源或执行线程接收一个或多个请求以创建所述阈值事件,接收与所述阈值事件相关联的多个条件,接收跟踪与所述阈值事件相关联的所述多个条件的至少一个功能,以及接收所述多个装置资源中的每一资源的所述唯一名称,所述多个装置资源将接收发生所述阈值事件的通知;

用于确定与所述未例示的节点相关联的所述多个装置资源中的每一资源是否可用于支持各自的请求的装置;以及

用于在所述未例示的节点相关联的所述多个装置资源中的任何资源不能用于支持各自的请求的情况下,组织其他装置资源使用所述未例示的节点的装置。

20. 根据权利要求19所述的系统,其中每一资源包括第二软件和第二硬件元件中的至少一者。

21. 根据权利要求19所述的系统,其进一步包括:

用于接收数据以用于形成充当所述节点框架中的至少一个功能资源的构造的占位符的存根资源的装置。

22. 根据权利要求21所述的系统,其进一步包括:

用于接收数据以用所述功能资源来取代存根资源的装置。

23. 根据权利要求19所述的系统,其进一步包括:

用于确定所述客户端请求是否可压制以延迟执行的装置。

24. 根据权利要求23所述的系统,其进一步包括:

用于确定是否已实现用于延迟可压制请求的执行的所述多个条件中的一个或多个的装置。

25. 根据权利要求19所述的系统,其中所述阈值事件包括所述便携式计算装置的温度阈值。

26. 根据权利要求25所述的系统,其中与所述阈值事件相关联的所述多个条件包括所述便携式计算装置的多个温度。

27. 根据权利要求19所述的系统,其中所述便携式计算装置包括移动电话、个人数字助理、寻呼机、智能电话、导航装置以及具有无线连接或链路的手持式计算机中的至少一者。

## 用于管理便携式计算装置的资源的方法和系统

### [0001] 相关申请案声明

[0002] 本专利申请案根据35 U.S.C. §120主张2010年9月15日申请的标题为“用于管理便携式计算装置的资源的方法和系统(SYSTEM AND METHOD FOR MANAGING RESOURCES OF A PORTABLE COMPUTING DEVICE)”的第12/882,395号美国专利申请案的优先权且是所述申请案的接续部分,所述申请案的说明书在此以全文引用的方式并入。本专利申请案还根据35 U.S.C. §119(e)主张2011年9月2日申请的标题为“用于管理便携式计算装置的资源的方法和系统(SYSTEM AND METHOD FOR MANAGING RESOURCES OF A PORTABLE COMPUTING DEVICE)”的第61/530,770号美国临时专利申请案的优先权,所述申请案的全部内容在此以引用的方式并入。

### 技术领域

#### 背景技术

[0003] 便携式计算装置(PCD)正在个人和专业层级上变为人们的必需品。这些装置可包含蜂窝式电话、便携式数字助理(PDA)、便携式游戏控制台、掌上型计算机,和其它便携式电子装置。这些装置中的每一者可包含主要功能。举例来说,蜂窝式电话一般具有接收和发射电话呼叫的主要功能。

[0004] 除了这些装置的主要功能之外,许多装置包含外围功能。举例来说,蜂窝式电话可包含如上文所描述的进行蜂窝式电话呼叫的主要功能,和静态相机、视频相机、全球定位系统(GPS)导航、网络浏览、发送和接收电子邮件、发送和接收文本消息、即按即说能力等的外围功能。随着此类装置的功能性增加,支持此类功能性所需的计算或处理能力也增加。此外,随着计算能力增加,更需要有效地管理提供计算能力的处理器。

[0005] 在过去,随着将由硬件或软件(或两者)支持的每一外围功能引入到例如蜂窝式电话等装置,针对每一外围功能引入专用编程接口(API)。举例来说,可存在用于摄像机的单独API以及用于GPS导航应用程序的单独API。每一API一般独立地记录其动作且每一API一般具有其自身的数据结构,所述数据结构将需要交叉参考在引入新的外围功能之前存在的蜂窝式电话的现有硬件或软件。

[0006] 针对每一外围功能引入单独的API是非常繁琐且耗时的,因为要交叉参考不同的硬件和软件元件。支持蜂窝式电话的基础功能的每一硬件或软件元件可已具备由蜂窝式电话的原始设备制造商(OEM)和/或支持蜂窝式电话的基础功能的基础电子器件的OEM确立的命名法。与软件或硬件(或两者)相关联的功能的新特征的记录和调试已长期被此便携式计算装置中的技术人员认为是在提供新产品或特征(或两者)的过程中的重大问题。

[0007] 所需要的是可以克服与引入新特征或功能相关联的问题的系统和方法,所述新特征或功能由添加到由原始设备制造商(OEM)建置的系统的新软件或硬件(或两者)支持。

### 发明内容

[0008] 本发明描述一种用于管理便携式计算装置的资源的方法和系统且其包含接收节点结构数据以用于形成节点,所述节点结构数据包括作为所述节点的每一资源的唯一名称。接下来,可针对一个或一个以上相依性来检视所述节点结构数据。随后确定与未例示的节点的相依性相关联的每一资源是否存在于节点框架内。

[0009] 所述方法和系统还可包含确定未例示的节点的每一资源是否可用于支持请求。如果未例示的节点的资源不可用于支持任何请求,那么可设定一值以防止包含不可用的资源的节点的例示。不可用的资源可包含可插拔的且可从PCD移除的硬件。

[0010] 如果与相依性相关联的资源完全不存在,那么可将所述节点结构数据存储于临时存储装置中。如果用于每一相依性的每一资源存在于节点结构数据内且可用,那么可创建或例示具有其一个或一个以上资源的节点。如果创建/例示所述节点,那么可使用对应于节点的处于准备好用于处理通信的状态中的一个或一个以上资源的一个或一个以上唯一名称在节点框架内公布所述节点。

[0011] 所述方法和系统可进一步包含确定例示的节点的每一资源是否可用于支持请求。如果例示的节点的资源不可用于支持任何请求,那么可锁定所述例示的节点而不可由其它资源接入。

[0012] 所述方法和系统可进一步包含接收数据以用于形成充当节点框架中的占位符的存根资源。在接收到存根数据之后,所述方法和系统可接收数据以用功能性资源来取代存根资源,使得可完成节点框架而仅包含功能性节点。

[0013] 所述方法和系统可进一步包含确定节点框架内所管理的客户端请求是否可压制以延迟执行,以便提高功率和/或性能。所述方法和系统可进一步包含确定是否已实现一个或一个以上条件来用于延迟可压制请求的执行。

[0014] 所述方法和系统可进一步包含接收一个或一个以上阈值事件以供资源管理。每一阈值事件可具有被所述资源跟踪的至少一个条件。

## 附图说明

[0015] 在各图中,相同的参考数字在所有各种视图指代相同的部分,除非另有指示。对于例如“102A”或“102B”等具有字母符号标示的参考数字,所述字母符号标示可区分同一图中存在的两个类似的部分或元件。当希望参考数字涵盖在所有图中具有相同参考数字的所有部分时,可省略参考数字的字母符号标示。

[0016] 图1是处于闭合位置的便携式计算装置(PCD)的第一方面的正视平面图;

[0017] 图2是处于打开位置的PCD的第一方面的正视平面图;

[0018] 图3是PCD的第二方面的框图;

[0019] 图4是处理系统的框图;

[0020] 图5是管理图1的便携式计算装置的资源的第一方面的图;

[0021] 图6A是提供管理图1的PCD的资源的第一方面的节点或资源图表的图;

[0022] 图6B是包含特定的又一示范性资源名称的节点的节点或资源图表的图;

[0023] 图6C是说明可相对于软件架构的资源进行跟踪的示范性温度阈值事件的图;

[0024] 图6D是说明资源的图表以及可由每一资源跟踪且可对应于图6C的温度阈值事件

的对应示范性温度事件的图；

[0025] 图7A是说明用于创建用于管理PCD的资源的软件架构的方法的流程图；

[0026] 图7B是说明用于创建用于管理PCD的资源的软件架构的方法的图7A的接续流程图；

[0027] 图7C是说明用于管理变为PCD不可用的未例示的节点的方法的图7A的接续流程图；

[0028] 图7D是说明用于管理PCD的可压制请求的方法的图7B的接续流程图；

[0029] 图7E是说明用于管理变为PCD不可用的例示的节点的方法的图7B的接续流程图；

[0030] 图8是说明用于在PCD中的软件架构中接收节点结构数据的图7A到B的子方法或例程的流程图；

[0031] 图9是说明用于在PCD的软件架构中创建节点的图7A到B的子方法或例程的流程图；

[0032] 图10是可由PCD的软件架构维持的示范性名称表的数据结构的图；

[0033] 图11是说明用于在PCD的软件架构中创建资源的别名的方法的流程图；

[0034] 图12是说明用于在PCD的软件架构中创建客户端的图9的子方法或例程的流程图；

[0035] 图13是说明用于在PCD的软件架构中创建针对资源的客户端请求的方法的流程图；

[0036] 图14是说明PCD的资源的等时客户端请求中所请求的工作的图；

[0037] 图15是说明用于在PCD的软件架构中创建针对资源的等时客户端请求的图9的子方法或例程的流程图；

[0038] 图16是说明用于在PCD的软件架构中创建针对资源的阈值事件的方法的流程图；  
以及

[0039] 图17是说明用于在PCD的软件架构中用新的(实际)资源取代存根资源的方法的流程图。

## 具体实施方式

[0040] 词语“示范性”在本文中用于表示充当实例、例子或说明。在本文中描述为“示范性”的任何方面不一定解释为比其它方面优选或有利。

[0041] 在此描述中,术语“应用”还可包含具有可执行内容的文件,例如:目标代码、脚本、字节代码、标记语言文件及补丁。另外,本文中所涉及的“应用”还可包含本质上不可执行的文件,例如可能需要打开的文档或需要存取的其它数据文件。

[0042] 术语“内容”还可包含具有可执行内容的文件,例如:目标代码、脚本、字节代码、标记语言文件及补丁。另外,本文中所提到的“内容”还可包含本质上不可执行的文件,例如可能需要打开的文档或其它需要存取的数据文件。

[0043] 如此描述中所使用,术语“组件”、“数据库”、“模块”、“系统”等意欲指代计算机相关实体,硬件、固件、硬件和软件的组合、软件或执行中的软件。举例来说,组件可为(但不限于为)在处理器上运行的进程、处理器、对象、可执行程序、执行线程、程序及/或计算机。以说明的方式,运行在计算装置上的应用程序和计算装置两者均可作为组件。一个或一个以上组件可驻留在进程和/或执行线程内,且组件可位于一个计算机上且/或分布在两个或两个



以上计算机之间。另外,这些组件可从上面存储有各种数据结构的各种计算机可读媒体来执行。所述组件可例如根据具有一个或一个以上数据包的信号(例如,来自一个借助于所述信号与本地系统、分布式系统中的另一组件交互及/或跨越例如因特网等网络而与其它系统交互的组件的数据)借助于本地及/或远程进程而通信。

[0044] 在此描述中,术语“通信装置”、“无线装置”、“无线电话”、“无线通信装置”和“无线手持机”可互换地使用。随着第三代(“3G”)无线技术和第四代(“4G”)无线技术的到来,更大的带宽可用性已实现具有更多种无线能力的更多便携式计算装置。因此,便携式计算装置可为蜂窝式电话、寻呼机、PDA、智能电话、导航装置或具有无线通信或链路的手持式计算机。

[0045] 起初参看图1和图2,展示示范性便携式计算装置(PCD)且大体上标示为100。如图所示,PCD 100可包含外壳102。外壳102可包含上部外壳部分104和下部外壳部分106(图2)。图1展示上部外壳部分104可包含显示器108。在特定方面中,显示器108可为触摸屏显示器。上部外壳部分104还可包含跟踪球输入装置110。此外,如图1中所示,上部外壳部分104可包含通电按钮112和断电按钮114。如图1中所示,PCD 100的上部外壳部分104可包含多个指示器灯116和扬声器118。每一指示器灯116可为发光二极管(LED)。

[0046] 在特定方面中,如图2中所描绘,上部外壳部分104可相对于下部外壳部分106移动。具体来说,上部外壳部分104可相对于下部外壳部分106滑动。如图2中所示,下部外壳部分106可包含多按钮键盘120。在特定方面中,多按钮键盘120可为标准QWERTY键盘。在上部外壳部分104相对于下部外壳部分106移动时可露出多按钮键盘120。图2进一步说明PCD 100可包含下部外壳部分106上的复位按钮122。

[0047] 参看图3,展示便携式计算装置(PCD)的示范性非限制性方面且大体上标示为100。如图所示,PCD 100包含芯片上系统322,所述芯片上系统包含多核CPU 402。多核CPU402可包含第零核410、第一核412和第N核414。

[0048] 如图3中所说明,显示器控制器328和触摸屏控制器330耦合到多核CPU 402。芯片上系统322外部的触摸屏显示器108又耦合到显示器控制器328和触摸屏控制器330。

[0049] 图3进一步说明视频编码器334(例如,逐行倒相(PAL)编码器、顺序传送彩色与存储(SECAM)编码器,或美国电视系统委员会(NTSC)编码器)耦合到多核CPU 402。此外,视频放大器336耦合到视频编码器334和触摸屏显示器108。而且,视频端口338耦合到视频放大器336。如图3中所描绘,通用串行总线(USB)控制器340耦合到多核CPU402。而且,USB端口342耦合到USB控制器340。存储器404和订户身份模块(SIM)卡346也可耦合到多核CPU 402。此外,如图3中所示,数码相机348可耦合到多核CPU402。在示范性方面中,数码相机348是电荷耦合装置(CCD)相机或互补金属氧化物半导体(CMOS)相机。

[0050] 如图3中进一步说明,立体声音频CODEC 350可耦合到多核CPU 402。此外,音频放大器352可耦合到立体声音频CODEC 350。在示范性方面中,第一立体声扬声器354和第二立体声扬声器356耦合到音频放大器352。图3展示麦克风放大器358也可耦合到立体声音频CODEC 350。此外,麦克风360可耦合到麦克风放大器358。在特定方面中,调频(FM)无线电调谐器362可耦合到立体声音频CODEC 350。而且,FM天线364耦合到FM无线电调谐器362。此外,立体声头戴式耳机366可耦合到立体声音频CODEC 350。

[0051] 图3进一步指示射频(RF)收发器368可耦合到多核CPU 402。RF开关370可耦合到RF

收发器368和RF天线372。如图3中所示,小键盘374可耦合到多核CPU 402。而且,具有麦克风的单声道耳机376可耦合到多核CPU 402。此外,振动器装置378可耦合到多核CPU 402。图3还展示电力供应器380可耦合到芯片上系统322。在特定方面中,电力供应器380为将电力提供给PCD 100的需要电力的各种组件的直流(DC)电力供应器。此外,在特定方面中,电力供应器为可再充电DC电池或从连接到AC电源的交流(AC)-DC变压器得到的DC电力供应器。

[0052] 图3进一步说明PCD 100还可包含网卡388,所述网卡可用于接入数据网络,例如局域网、个域网或任何其它网络。网卡388可为蓝牙网卡、WiFi网卡、个域网(PAN)卡、个域网超低功率技术(PeANUT)网卡,或此项技术中众所周知的任何其它网卡。此外,网卡388可并入到芯片中,即网卡388可为芯片上的整个解决方案,且可不为单独的网卡388。

[0053] 如图3中所描绘,触摸屏显示器108、视频端口338、USB端口342、相机348、第一立体声扬声器354、第二立体声扬声器356、麦克风360、FM天线364、立体声头戴式耳机366、RF开关370、RF天线372、小键盘374、单声道耳机376、振动器378,和电力供应器380在系统芯片322的外部。

[0054] 在特定方面中,本文中所描述的方法步骤中的一者或一者以上可作为计算机程序指令存储于存储器404中。这些指令可由多核CPU 402执行以便执行本文中所描述的方法。此外,多核CPU 402、存储器404或其组合可作用于执行本文中所描述的方法步骤中的一者或一者以上以便在中央处理单元402内对数据进行取样的装置。

[0055] 参看图4,展示处理系统且大体上标示为400。在特定方面中,处理系统400可并入到上文结合图3所描述的PCD 100中。如图所示,处理器系统400可包含多核中央处理单元(CPU) 402和连接到多核CPU 402的存储器404。多核CPU 402可包含第零核410、第一核412和第N核414。第零核410可包含在其上执行的第零动态时钟和电压缩放(DCVS)算法416。第一核412可包含在其上执行的第一DCVS算法417。此外,第N核414可包含在其上执行的第N DCVS算法418。在特定方面中,每一DCVS算法416、417、418可独立地在相应核412、414、416上执行。

[0056] 另外,如所说明,存储器404可包含存储于其上的操作系统420。操作系统420可包含总线仲裁器或调度器422,且调度器422可包含第一运行队列424、第二运行队列426,和第N运行队列428。存储器404还可包含存储于其上的第一应用430、第二应用432和第N应用434。

[0057] 在特定方面中,应用430、432、434可将一个或一个以上任务436发送到操作系统420以在多核CPU 402内的核410、412、414处进行处理。可将任务436处理或执行为单个任务、线程或其组合。此外,调度器422可调度任务、线程或其组合以用于在多核CPU 402内执行。另外,调度器422可将任务、线程或其组合放置于运行队列424、426、428中。核410、412、414可如(例如)操作系统420所指令而从运行队列424、426、428检索任务、线程或其组合,以用于在核410、412、414处对那些任务和线程的处理或执行。

[0058] 图4还展示存储器404可包含存储于其上的框架管理器440。框架管理器440可连接到操作系统420和多核CPU 402。具体来说,框架管理器440可连接到操作系统420内的调度器422。如本文中所描述,框架管理器440可监视核410、412、414上的工作负荷,且框架管理器440可对来自核410、412、414的数据进行取样。

[0059] 在特定方面中,框架管理器440可为软件程序。然而,在替代性方面中,框架管理器

440可为在存储器404外部的硬件控制器。在任何情况下,框架管理器440、存储器404、核410、412、414或其任何组合可作用于执行本文中所描述的方法步骤中的一者或一者以上以便对来自核410、412、414的数据进行取样的装置。

[0060] 图5是管理图1的便携式计算装置(PCD)的资源的系统的软件架构500A的第一方面的图。图5是包括表示软件或硬件(或两者)的功能块的图。图5说明耦合到多个硬件和软件元件的架构或框架管理器440,所述硬件和软件元件例如为(但不限于):中央处理单元402,还一般被称作第一硬件元件(硬件元件#1);用于CPU 402的时钟442,还一般被称作第二硬件元件(硬件元件#2);总线仲裁器或调度器422,还一般被称作第三硬件元件(硬件元件#3);总线程序A-444A,还一般被称作第一软件元件(软件元件#1);总线程序B-444B,还一般被称作第二软件元件(软件元件#2);时钟程序AHB,一般被称作第三软件元件(软件元件#3);被一般指示为按键448的软件元件监视的动作或功能;以及传统元件450,其包括软件元件或软件元件或两者。

[0061] 传统软件元件的实例可包含(但不限于)动态环境管理器(“DEM”)。这是处置处理器休眠事件的处理器间通知的软件模块。举例来说,第一处理器A使用DEM来接收第二处理器B已经变为闲置/从闲置回来的通知。在更新的硬件上,此软件功能性已被归入路由处理器模块(“RPM”)子系统/通信协议中。存在其它传统软件元件且包含于本发明的范围内。

[0062] 传统硬件元件的实例可包含(但不限于)AMBA(“高级控制器总线架构”)高性能总线(“AHB”)。在较老的PCD 100上。AHB可包括主要系统总线,而在较新的PCD 100上,系统总线构造完全不同,且AHB总线仅用于与尚未经更新以经由新的系统总线构造进行通信的模块通信的特殊应用。存在其它传统硬件元件且包含于本发明的范围内。

[0063] 框架管理器440可包括计算机指令库,其管理与上述硬件元件和软件元件中的每一者进行通信的数据结构,例如代码(下文描述)。框架管理器440可负责创建一个或一个以上资源,所述资源可形成如在图5的虚线A的右侧上说明的节点602、622、642和646。框架管理器可包括代码库。所述库的若干部分可驻留于PCD 100的每一硬件元件上。

[0064] 同时,由框架管理器440建立和例示的每一节点602、622、642和646是在图5的虚线A的左手侧上的每一软件或硬件元件的表示或模型。对于本发明的剩余部分,将用如图6A中所说明的参考数字601来标示一般或非特定节点。

[0065] 如先前所述,图5的每一示范性节点602、622、642和646可包括一个或一个以上资源。资源可包括软件元件或硬件元件或两者。举例来说,第一节点602包括一般与第一硬件元件或中央处理单元402对应的单个资源。通过本发明中所描述的发明性软件架构,节点601的每一资源可具备包括一个或一个以上字母数字字符的唯一名称。在图5中所说明的示范性实施例中,已向第一节点602的资源指派资源名称“/core/cpu”。此示范性资源名称一般对应于所属领域的技术人员已知的常规文件命名结构。然而,如所属领域的技术人员所认识到的,含有字母数字字符和/或符号的任何其它组合的其它类型的资源名称也在本发明的范围内。

[0066] 在图5的示范性实施例中,第二节点622包括多个资源。特定来说,在此特定示范性实施例中,第二节点622具有包括对应于总线仲裁器或调度器422的单个硬件元件的第一资源。第二节点622的第二资源包括一般对应于总线程序A 444A的第一软件元件的软件元件。第二节点622的第三资源包括一般对应于总线程序B 444B的第二软件元件的另一软件元

件。所属领域的技术人员应认识到,任何组合和任何数目的用于给定节点601的资源和资源类型也在本发明的范围内。

[0067] 图5还说明一般对应于两个软件元件448、450的动作或功能的第一客户端648。在图5中所说明的示范性实施例中,客户端648一般对应于可在由便携式计算装置100支持的特定应用程序内发生的按键动作。然而,所属领域的技术人员应认识到,除了按键之外的软件元件的其它动作和/或功能也在本发明的范围内。下文将结合图12描述关于客户端648及其相应创建的进一步细节。

[0068] 图5还说明特定架构性元件之间的关系。举例来说,图5说明客户端648与第一节点602之间的关系。具体来说,第一客户端648可产生用虚线说明的客户端请求675A,所述客户端请求由包括资源“/core/cpu”的第一节点602管理或处置。通常,存在预定数目或设定数目的类型的客户端请求675。下文将结合图13进一步详细地描述客户端请求675。

[0069] 图5中所显示的其它关系包含用虚线680说明的相依性。相依性是另一节点601的相应资源之间的关系。相依性关系通常指示第一资源(A)依赖于可向第一资源(A)提供信息的第二资源(B)。此信息可为由第二资源(B)执行的操作的结果,或其可简单地包括第一资源(A)所需的状态信息,或其任何组合。第一资源(A)和第二资源(B)可为相同节点601的部分,或其可为不同节点601的部分。

[0070] 在图5中,第一节点602取决于第二节点622,如由以第一节点602起始且延伸到622处的第二节点的相依性箭头680B所指示。图5还说明第一节点602还取决于第三节点642,如由相依性箭头680A所说明。图5还说明第二节点622取决于第四节点646,如由相依性箭头680C所说明。所属领域的技术人员认识到,用图5的虚线箭头所说明的相依性680在本质上仅是示范性的,且相应节点601之间的相依性的其它组合在本发明的范围内。

[0071] 框架管理器440的架构负责维持上文所描述的关系,其包含(但不限于)图5中所说明的客户端请求675和相依性680。只要用于任何给定节点601的相依性680是完整的,框架管理器440将尝试例示或创建尽可能多的节点601。当支持相依性的资源存在或处于用于处置与相依性680相关的信息就绪状态时,相依性680便完整。

[0072] 举例来说,由于在第一节点602与第三节点642之间存在的相依性关系680A,如果包括单个资源“/clk/cpu”的第三节点642尚未创建,那么包括单个资源“/core/cpu”的第一节点602可不会被框架管理器440创建或建立。一旦第三节点642已被框架管理器440创建,那么框架管理器440可由于相依性关系680A而创建第二节点602。

[0073] 如果框架管理器440无法创建或例示特定节点601(因为其相依性680中的一者或一者以上不完整),那么框架管理器440将继续运行或执行对应于曾由框架管理器440成功创建的那些节点601的步骤。框架管理器440将通常跳过对可由于不完整相依性(其中尚未创建相依资源)而不存在的特定节点601的调用,且将反映所述不完整状态的消息返回到所述调用。

[0074] 在例如图4中所说明的多核环境中,框架管理器440可在比如图4的第一、第二和第N核424、426和428的单独核上创建或例示节点601。只要节点601彼此不相依且在特定节点的对应相依性(下文描述)全部完整的情况下,节点601可一般在单独核上的多核环境中创建。

[0075] 图6A是用于管理图1的PCD 100的资源的系统的软件架构500B1的第二方面的一般

图。在此一般图中，每一节点601的一个或一个以上资源尚未具备唯一名称。图6A的节点或资源图表500B1仅包括节点601、客户端648、事件690和由所述架构或框架管理器440支持的查询功能695。每一节点601已用椭圆形状和具有表示节点601内的资源之间的相应相依性的特定方向的箭头680来说明。

[0076] 可对节点601内的资源的别名或实际资源名称作出图6A到B中所说明的节点架构内的调用。举例来说，第一节点601A具有相依性箭头680A以指示第一节点601A取决于第二节点601B的两个资源(资源#2和#3)。图6A还说明第一节点601A的客户端648可如何向第一节点601A发出客户端请求675。在发出这些客户端请求675之后，第二节点601B可触发事件690或将响应提供给查询695，其中对应于事件690和查询695的消息流回到客户端648。

[0077] 图6B是用于管理图1的PCD 100的资源的系统的软件架构500B2的第二方面的特定图。图6B说明节点或资源图表500B2，其仅包括具有特定的又一示范性资源名称的节点601，以及对应于图5的对应物的客户端648、事件690和查询功能695。每一节点601已用椭圆形状和具有表示节点601内的资源之间的相应相依性的特定方向的箭头680来说明。

[0078] 举例来说，第一节点602具有相依性箭头680B以指示第一节点602取决于第二节点622的三个资源。类似地，包括第二软件元件444B且一般用图6中的参考字母“C”标示的第三资源“/bus/ahb/sysB/”具有相依性箭头680C，所述相依性箭头指示此第三资源(C)取决于第四节点646的单个“/clk/sys/ahb”资源。

[0079] 图6B还说明来自可包括一个或一个以上事件690或查询功能695的节点601的输出数据。查询功能695类似于事件690。查询功能695可具有可为或可不为唯一的查询句柄。

[0080] 查询功能一般不在外部识别，且其一般不具有状态。查询功能695可用于确定节点601的特定资源的状态。查询功能695和事件690可与所建立的客户端648具有关系，且这些关系可由方向性箭头697指示，以指示来自相应事件690和查询功能695的信息被传递到特定客户端648。

[0081] 图6的节点或资源图表500B表示例如图4的存储器404等存储器中所存在的关系，且所述关系由框架管理器440和可包括节点601的相关数据结构管理。节点或资源图表500B可由框架管理器440自动产生为用于识别由框架管理器440管理的相应元件之间的关系且用于通过软件小组来排除故障的有用工具。

[0082] 图6C是说明可相对于软件架构500的资源进行跟踪的示范性温度阈值事件690的图。虽然图6C中所说明的阈值事件690与温度相关，但所属领域的技术人员应认识到，阈值事件可与可在PCD 100内监视的任何类型的条件相关联。举例来说，其它阈值事件包含(但不限于)与电力消耗、电池电荷状态/状况、休眠或低电力条件、特定硬件条件(例如，良好、好和差射频(“RF”)接收)相关的阈值事件、可与比如每秒百万指令(“MIP”)、时钟频率、总线频率、电压、DCVS条件和算法等负载相关的操作条件。

[0083] 阈值事件690通常包含一个或一个以上触发条件。在图6C中所说明的实例中，触发条件可包括可用PCD 100测量的一个或一个以上温度。在图6C的实例中，已选择三个阈值温度：A)可表示PCD 100的高热状态的八十摄氏度；B)可表示PCD 100的标称热状态的六十五摄氏度；以及C)可表示PCD 100的低热状态的五十摄氏度。

[0084] 当温度处于五十摄氏度或以下时，已创建第一阈值事件690A以跟踪热条件。当温度是五十摄氏度或以下时或当温度是八十摄氏度或以上时，已创建第二阈值事件690B以跟

踪热条件。当温度是六十五摄氏度或以下时,已创建第三阈值事件690C以跟踪热条件。且当温度是五十摄氏度或以下时或当已达到六十五摄氏度时或当已达到八十摄氏度时,已创建第四阈值事件690D以跟踪热条件。

[0085] 可将这四个不同阈值事件690A、690B、690C、690D中的每一者指派给PCD 100的资源的不同、相同或各种组合,如下文结合图6D进一步详细地描述。由框架管理器440支持的软件架构500的任何用户可针对资源设置阈值事件690。创建阈值事件690的软件架构500的用户可包含资源创造者、节点创造者、资源,或执行线程。可指令指派给阈值事件690的资源向一个或一个以上不同资源报告阈值事件的状态。

[0086] 图6D是说明资源的图表697以及可由每一资源跟踪的对应示范性温度阈值事件690的图。图6D的这些温度阈值事件690对应于图6C的温度阈值事件690。根据此示范性实施例,图表697的第一列列举了负责跟踪第二列中所列举的阈值事件的每一资源。第三列列举了将在达到阈值时接收阈值事件的状态的资源接收方。

[0087] 已向图6A的第一资源(资源#1)指派两个温度阈值事件:图6C的第一温度阈值事件690A和第二温度阈值事件690B。第一资源将把其阈值状态传输到单个资源:第四资源。已向图6A的第二资源(资源#2)指派了两个温度阈值事件:图6C的第一温度阈值事件690A和第三温度阈值事件690C。第二资源将把其阈值状态报告给单个资源:第五资源。且已向图6A的第三资源(资源#3)指派了单个温度阈值事件:图6C的第四阈值温度事件690D。第三资源将把其阈值状态报告给两个资源:第四资源和第六资源。

[0088] 图6D的图表697说明可如何将阈值事件690指派给资源以及可如何报告这些事件690的灵活性和缩放性。框架管理器440负责监视并报告图表697中所列举的资源之间的通信。下文将结合图16来描述用于创建阈值事件690的方法或过程1600。用于创建阈值事件690的方法1600可在用于创建软件架构500以及用于管理PCD 100的资源的方法700A已被执行之后执行。

[0089] 图7A是说明用于创建软件架构以及用于管理PCD 100的资源的方法700A的流程图。框705是用于管理PCD 100的资源的方法或过程700的第一例程。在框705中,可由框架管理器440执行或运行例程来接收节点结构数据。所述节点结构数据可包括相依性阵列,所述相依性阵列概述了特定节点601与其它节点601可具有的相依性。下文将结合图8更详细地描述关于节点结构数据和此例程或子方法705的进一步细节。

[0090] 接下来,在框710中,框架管理器440可检视作为框705中所接收的节点结构数据的部分的相依性数据。在决策框715中,框架管理器440可确定节点结构数据是否界定叶节点601。叶节点601一般是指将基于节点结构数据创建的节点不具有任何相依性。如果对决策框715的询问是肯定的,意味着用于创建当前节点的节点结构数据不具有任何相依性,那么框架管理器440继续进行到例程框725。

[0091] 如果对决策框715的询问是否定的,那么遵循“否”分支到决策框720,其中框架管理器确定节点结构内的全部硬相依性是否存在。硬相依性可包括其中资源没有它就无法存在的一者。同时,软相依性可包括其中资源可将相依资源用作任选步骤的一者。软相依性意味着可在节点架构内创建或例示具有软相依性的节点601或节点601的资源,甚至在软相依性不存在时也如此。

[0092] 软相依性的实例可包括对于含有多个资源的面向资源的节点601的操作不重要的

优化特征。框架管理器440可创建或例示用于所存在的所有硬相依性的节点或资源,且甚至在软相依性对于具有不存在的软相依性的那些节点或资源不存在时也是如此。可使用回调特征来参考软相依性,使得当软相依性变得可用于框架管理器440时,框架管理器440将向参考软相依性的每一回调查告知软相依性现在可用。

[0093] 如果对决策框720的询问是否定的,那么遵循“否”分支到框731,其中框架管理器440将节点结构数据存储在例如存储器等临时存储装置中。框架管理器440创建与此未例示的节点相关联的回调特征。在框731之前且在决策框720的否定条件或肯定条件之后,可执行用于管理变为不可用的未例示的节点601的任选方法700C。已经用图7A中的框720与框731之间的虚线来说明此任选方法700C。下文将结合图7C描述任选方法700C的进一步细节。

[0094] 如果对决策框715的询问是肯定,那么遵循“是”分支到框725,其中基于在例程框705中接收的节点结构数据来创建或例示节点601。下文将结合图9描述例程框725的进一步细节。接下来,在框732中,框架管理器440使用其唯一的资源名称来公布新创建的节点601,使得其它节点601可将信息发送到新创建的节点601或从所述节点接收信息。

[0095] 现在参看作为图7A的接续流程图的图7B,在框735中,框架管理器440通知取决于新创建的节点601的其它节点601新创建的节点601已被例示且准备好接收或传输信息。

[0096] 根据一个示范性方面,在创建相依节点(比如图6A的节点601B)时立即触发通知,即,递归地执行通知。因此在建构图6A的节点601B的情况下,立即通知节点601A。此通知可允许建构节点601A(因为节点601B是节点601A的最终相依性)。节点601B的构造可致使其它节点601被通知,且以此类推。节点601B直到取决于节点601B的最终资源完成时才完成。

[0097] 第二、稍微更复杂的实施方案是将全部通知放到单独的通知队列上,且随后在单个时间点处行进穿过所述队列,即,递归地执行通知。因此在建构图6A的节点601B时,将对节点601A的通知推到列表上。随后执行所述列表且通知节点601A。此致使对其它额外节点601(除了节点601A之外,图6A中未说明)的通知被放在相同列表上,且随后在发送对节点601A的通知之后发送所述通知。对其它节点601的通知(除了对节点601A的通知之外)直到在已完成与节点601B和节点601A相关联的所有工作之后才发生。

[0098] 逻辑上,这两个实施方案是确切等效的,但其在实施时具有不同的存储器消耗性质。递归实现是简单的,但可消耗任意量的堆栈空间,其中堆栈消耗随相依性图表的深度而变。迭代实施方案稍微更复杂且需要多一点的静态存储器(通知列表),但堆栈使用是恒定的,而不管相依性图表的深度如何,例如图6A中所说明。

[0099] 而且,框735中的节点创建的通知不限于其它节点。其还可在内部用于别名建构。系统500中的任何任意元件可使用相同机制以在节点(或标记)变得不可用(不仅仅是其它节点)时请求通知。节点和非节点两者可使用相同的通知机制。

[0100] 在决策框740中,框架管理器440确定现在是否释放其它节点601或软相依性以用于基于当前节点601的创建进行创建或例示。决策框740一般确定现在是否可创建资源,因为已由近来已经受创建或例示的当前节点完成某些相依性关系680。

[0101] 如果对决策框740的询问是肯定的,那么遵循“是”分支回到例程框725,其中因为刚创建的节点601完成相依性,所以现在可创建或例示经释放的节点601。

[0102] 如果对决策框740的询问是否定的,那么遵循“否”分支到框741,其中框架管理器440可管理如图5和6中所说明的软件架构的元件之间的通信。在框741与框753之间,可通过

框架管理器440和资源来执行用于管理可压制请求的任选方法700D。下文将结合图7D描述任选方法700D的进一步细节。

[0103] 接下来,在框753中且在任选方法700D之后,框架管理器440可通过使用与特定资源相关联的资源名称来继续记录或记载由资源采取的动作。框753可由框架管理器440在框架管理器440或框架管理器440所管理的元件(例如,资源、节点601、客户端648、事件695和查询功能697)中的任一者所采取的任何动作之后执行。框753是本发明的又一个重要方面,其中框架管理器440可维持活动的运行日志,所述日志列举了由每一元件根据其由创建了特定元件(例如,节点601的资源)的创造者提供的唯一识别符或名称而执行的动作。

[0104] 与现有技术相比,框753中的列举了指派给系统的每一资源的唯一名称的此活动日志是唯一的,且可提供显著优点,例如用于调试和错误排除故障中。使系统500唯一的许多方面中的另一方面是单独小组可对不同的硬件和/或软件元件彼此独立地工作,其中每一小组将能够使用唯一的且容易跟踪而不需要创建表的资源名称来翻译由其它小组和/或原始设备制造商(OEM)指派的含义较少且通常混淆的资源名称。

[0105] 接下来,在决策框755中,框架管理器440确定是否已请求框架管理器440所记录的活动日志。如果对决策框755的询问是否定的,那么遵循“否”分支到过程结束,其中过程返回到例程705。如果对决策框755的询问是肯定的,那么遵循“是”分支到框760,其中框架管理器440将包括有含义的资源名称以及由资源名称执行的相应动作的活动日志发送到输出装置,例如打印机或显示屏和/或两者。变得不可用的用于管理未例示的节点601的任选方法700E可在框760之后执行。下文将结合图7D描述方法700E的进一步细节。在框760之后,过程700B可随后返回到上文所描述的例程框705。

[0106] 图7C是说明用于管理变为PCD 100不可用的未例示的节点601的任选方法700C的图7A的接续流程图。当节点601的资源离线或已经历故障条件时,可将节点601表征为不可用。根据一个示范性方面,如果资源是可从PCD 100移除的硬件(例如装置的USB端口类型),那么所述资源可变得不可用。当硬件资源已经从PCD 100移除且不可用于处理可由框架管理器440管理的任何请求时,硬件资源的驱动器可告知图5的框架管理器440。

[0107] 此任选方法700C可良好适合于可耦合到PCD 100的可插拔/不可插拔类型的资源。此任选方法700C还针对在图7A的例程框725中尚未例示或创建的节点601进行裁剪。换句话说,任选方法700C针对尚未由框架管理器440创建的节点601来管理节点结构数据,其中创建/例示通常发生在图7A的例程框725中。当对应于节点的资源因为其未从系统500插拔或因为其正在经历一个或一个以上故障条件而不可用时,任选方法700C阻止创建节点601。

[0108] 决策框721是任选方法700C的第一步骤。在此决策框721中,图5的框架管理器440可为监视消息,其可从可用于系统500的可插拔资源的装置驱动器接收。如先前所述,除了管理可用于系统500的可插拔资源之外,任选方法700C还可管理已经历故障条件且阻止资源取得或作用于由框架管理器440发送到其的任何请求的那些资源。资源的装置驱动器可在决策框721中告知框架管理器440节点601的特定资源是否已经历需要框架管理器442使所述资源及其对应节点601离线的故障条件。

[0109] 如果对决策框721的询问是否定的,那么遵循“否”分支到框722,其中过程返回到图7A的框731(如果从决策框720遵循“否”分支)或过程返回到例程框725(如果从决策框720遵循“是”分支)。如果对图7C的决策框721的询问是肯定的,那么遵循“是”分支到框723。



[0110] 在框723中, 框架管理器440锁定未例示的节点601而不让一个或一个以上其它资源存取以作为独立节点601。换句话说, 当未例示的节点601的一个或一个以上资源不可让系统500用于处理由其它节点601、线程、客户端648等发出的请求时, 框架管理器440阻止其它节点601参考所述未例示的节点。

[0111] 在框724中, 框架管理器440停用未例示的节点601以防止其由框架管理器440稍后例示或创建。换句话说, 在此框724中, 框架管理器440可设定由框架管理器440跟踪的防止未例示的节点601变为图6A的框架或资源图表500B1的部分的旗标或值, 直到形成未例示的节点601的所述资源变得可用于取得来自系统500的请求为止。

[0112] 在决策框726中, 框架管理器440用未例示的节点601的资源的一个或一个以上装置驱动器进行检查以确定不可用的一个或一个以上资源是否已改变状态而变为可用于系统500。如果对决策框726的询问是否定的, 那么遵循“否”分支到框730, 其中过程700返回到图7A的框731。

[0113] 如果对决策框726的询问是肯定的, 那么遵循“是”分支到框727, 其中框架管理器440启用未例示的节点601, 使得当已开始图7A的例程725时, 可例示或创建所述节点。在此框727中, 框架管理器440可清除或复位与未例示的节点601相关联的旗标, 使得可由框架管理器440在图7A的例示例程725期间处理未例示的节点601。

[0114] 在决策框728中, 框架管理器440可确定是否还启用又重新启用的未例示的节点601的相依性。此决策框728允许框架管理器442确保又重新启用的未例示的节点601所相依的节点601或节点601的资源自身已启用。如果对决策框728的询问是否定的, 那么可遵循“否”分支回到框730, 其中过程700返回到图7A的框731 (如果从决策框720到遵循“否”分支到决策框721) 或过程返回到例程框725 (如果从决策框720遵循“是”分支到决策框721)。

[0115] 如果对图7C的决策框728的询问是肯定的, 那么遵循“是”分支到框729, 其中框架管理器440解开重新启用的未例示的节点601。当在执行图7A的例示例程725之后产生图6A的资源图表500B1时, 重新启用的未例示的节点601经解开以让其它资源存取。任选方法700C随后返回到框731 (如果从决策框720到遵循“否”分支到决策框721) 或过程返回到例程框725 (如果从决策框720到遵循“是”分支到决策框721)。

[0116] 图7D是说明用于管理由客户端、资源和/或线程等发出的可压制请求的任选方法700D的图7A的继续流程图。可压制请求是当已满足特定资源的条件或阈值时可忽视资源或将资源搁置的请求。所述阈值或条件可经建立且针对每一资源可为唯一的。示范性阈值或条件可包含 (但不限于) 休眠状态、高热条件、低电池电力条件以及差无线电接收等。可压制请求通常被识别为在如下文结合图13所描述的方法1300A (例如在步骤1305中) 期间创建之后的请求。

[0117] 可使用在创建请求期间由客户端设定的旗标或变量来识别可压制请求。资源可选择利用可压制请求特征来管理其请求, 或其可选择忽略是否是否可压制。

[0118] 当资源确定在给定情景下中止请求或搁置请求可导致PCD 100的更大效率时, 可压制请求对于资源来说是有利的。举例来说, 假设资源负责管理总线。所述资源还可在耦合到总线的CPU想要进入闲置状态中时学习条件。

[0119] 所述资源可接收用以增加正由总线支持的带宽的请求。同时, 资源可已刚刚接收到CPU将要进入闲置状态中的消息。如果所述资源知道对带宽增加的总线请求是可压制请

求,那么所述资源可能够压制所述总线请求,直到CPU退出其闲置状态为止。让资源管理总线以在CPU处于闲置状态中时压制总线请求的此能力给予PCD 100的显著效率以便节省电力。所属领域的技术人员鉴于此实例会容易理解用于用可压制请求来增加管理资源的效率的其它示范性情景。

[0120] 决策框742是用于管理可压制请求的方法700D的第一步骤。在决策框742中,节点601的资源可确定其当前是否正在处置一个或一个以上被压制的请求。如果对决策步骤742的询问是否定的,那么遵循“否”分支到决策框745。

[0121] 如果对决策框742的询问是肯定的,那么遵循“是”分支到决策框743。在决策框743中,资源确定是否存在用于执行先前被压制的一个或一个以上请求的一个或一个以上条件。举例来说,在上文关于CPU 402进入闲置状态中所述的示范性情景中,所述资源可确定CPU 402是否已离开其闲置状态以作为一条件。如果此条件是真,意味着CPU402已离开其闲置状态,那么所述资源可执行先前被压制的请求。换句话说,如果对决策框743的询问是肯定的,那么可遵循“是”分支到框744,其中资源执行先前被压制的一个或一个以上请求。实质上,此点处的代码计算在压制条件变为真(即,如果CPU 402变闲置)时新状态应为什么且将其存储起来。随后在所述条件变为真时,其应用另一状态。这是延迟执行的形式,如所属领域的技术人员所理解。

[0122] 如果对决策框743的询问是否定的,那么遵循“否”分支到决策框745。在决策框745中,资源可确定是否存在用于压制已被识别为可压制的那些请求的一个或一个以上条件。决策框745可与其中CPU将要进入闲置状态中的先前实例相对应。如果对决策框745的询问是肯定的,那么可遵循“是”分支到框746,其中资源可聚集一个或一个以上可压制请求以供资源稍后执行。如果对决策框745的询问是否定的,那么可遵循“否”分支到框747,其中过程700D返回到图7B的框753。

[0123] 图7E是说明用于管理变为PCD 100不可用的未例示的节点601的任选方法700E的图7A的接续流程图。此任选方法700E非常类似于任选方法700C,除了任选方法700E是用于已基于图7A的例程725例示的节点601之外。

[0124] 如先前所述,当节点的资源离线或已经历故障条件时,可将节点601表征为不可用。根据一个示范性方面,如果资源是可从PCD 100移除的硬件(例如装置的USB端口类型),那么所述资源可变得不可用。当资源已经从PCD 100移除且不可用于处理可由框架管理器440管理的任何请求时,硬件的驱动器可告知图5的框架管理器440。

[0125] 此任选方法700E与任选方法700C一样可良好适合于可耦合到PCD 100的可插拔/不可插拔类型的资源。此任选方法700E针对在图7A的例程框725中已例示或创建的节点601进行裁剪。换句话说,任选方法700E的节点601已由框架管理器440创建且已存在于例如图6A的资源图表500B1等资源图表中。

[0126] 决策框766与图7A的框721一样是任选方法700D的第一步骤。在决策框766中,框架管理器440可确定是否应使如图6A的示范性节点资源图表500B1中所说明的特定节点601离线。在此决策框766中,图5的框架管理器440可为监视消息,其可从可用于系统500的可插拔资源的装置驱动器接收。如先前所述,除了管理可用于系统500的可插拔资源之外,任选方法700E还可管理已经历故障条件且阻止资源取得或作用于由框架管理器440发送到其的任何请求的那些资源。资源的装置驱动器可在决策框766中告知框架管理器440节点601的特

定资源是否已经历需要框架管理器442使所述资源及其对应节点601离线的故障条件。

[0127] 如果对决策框766的询问是否定的,那么遵循“否”分支回到图7A的框705。如果对决策框766的询问是肯定的,那么遵循“是”分支到框768。

[0128] 在框768中,框架管理器440在使节点601离线之前记录节点601的资源的当前状态。框架管理器440可随后使资源及其对应的节点601离线。

[0129] 在框770中,框架管理器440锁定现有节点601而不让一个或一个以上其它资源存取以作为独立节点601。换句话说,当节点601的一个或一个以上资源不可让系统500用于处理由其它节点601、线程、客户端648等发出的请求时,框架管理器440阻止其它节点601参考所述节点。

[0130] 在框772中,框架管理器440停用节点601及其当前状态。在决策框774中,框架管理器440用经停用的节点601的资源的一个或一个以上装置驱动器进行检查以确定不可用的一个或一个以上资源是否已改变状态而变为可用于系统500。

[0131] 如果对决策框774的询问是否定的,那么遵循“否”分支回到图7A的框705。如果对决策框774的询问是肯定的,那么遵循“是”分支到框776,其中框架管理器440启用节点601,使得其可再次用于系统500。在框778中,重新初始化现有节点601,使得将其带回其前一状态。换句话说,重新初始化现有节点601以便将节点601置于恰在框架管理器440使其离线之前的状态中。

[0132] 在决策框780中,框架管理器440可确定是否还启用重新启用的节点601的相依性。此决策框728允许框架管理器442确保重新启用的现有节点601所相依的节点601或节点601的资源自身已启用。如果对决策框780的询问是否定的,那么可遵循“否”分支回到图7A的框705。

[0133] 如果对决策框780的询问是肯定的,那么遵循“是”分支到框782,其中框架管理器440解开重新启用的现有节点601。重新启用的现有节点601被解开以让其它资源存取,如图6A的资源图表500B1中所说明。任选方法700E随后返回到图7A的框705。

[0134] 图8是说明用于在PCD 100的软件架构中接收节点结构数据的图7的子方法或例程705的流程图。决策框802是图8的子方法或例程705中的第一步骤。

[0135] 在决策框802中,框架管理器440确定节点结构数据是否含有用于创建存根资源的请求。存根资源是允许资源创造者在图6A的框架或资源图表500B1中创建占位符的软件开发工具。存根资源是可允许形成其它资源且其它资源可相依的占位符。以此方式,当特定资源不可用或尚未开发出时,可由框架管理器440在例示例程725期间创建其它资源和最终的节点和客户端。

[0136] 节点的其它资源可登记针对于存根资源的事件。然而,存根资源将从不返回事件的值,因为存根资源将不含有可支持事件的任何有效功能或特征。可用稍后阶段处的实际或运行资源来取代存根资源。下文将结合图17来描述关于可如何用实际资源取代存根资源的进一步细节。

[0137] 如果对决策框802的询问是否定的,那么遵循“否”分支到框805。如果对决策框802的询问是肯定的,那么遵循“是”分支到框803,其中框架管理器440可接收用于创建存根资源的详细名称和数据。过程705随后返回到图7A的框710。

[0138] 在框805中,框架管理器440可接收例如图5的CPU 402和时钟442等软件或硬件元

件的唯一名称。如先前所论述,节点601必须参考至少一个资源。每一资源具有一名称,且所述名称在系统500中必须是唯一的。可用唯一名称来识别系统500内的所有元件。每一元件从字符角度来看具有唯一名称。换句话说,一般来说,在系统500内不存在具有相同名称的两个元件。

[0139] 根据系统的示范性方面,节点601的资源可一般在系统上具有唯一名称,但不需要客户端或事件名称是唯一的,但它们可在需要时是唯一的。

[0140] 为了方便起见,可使用使用正斜杠“/”字符来用于创建唯一名称的常规树文件命名结构或文件命名“隐喻”,例如但不限于用于CPU 402的“/core/cpu”和用于时钟442的“/clk/cpu”。然而,如所属领域的技术人员所认识到的,含有字母数字字符和/或符号的任何其它组合的其它类型的资源名称也在本发明的范围内。

[0141] 接下来,在框810中,框架管理器440可接收用于与正被创建的节点601的一个或一个以上资源相关联的一个或一个以上驱动器功能的数据。驱动器功能一般包括将由特定节点601的一个或一个以上资源完成的动作。举例来说,在图6中,节点602的资源/core/cpu的驱动器功能可请求其所需的总线带宽的量和CPU时钟频率以便提供已请求的所请求的处理量。这些请求将经由节点642和节点622中的资源的客户端(未说明)来作出。节点642中的/clk/cpu的驱动器功能将通常负责根据其从节点602的/core/cpu资源接收到的请求来实际设定物理时钟频率。

[0142] 在框815中,框架管理器440可接收节点属性数据。节点属性数据一般包括界定节点策略的数据,所述节点策略例如为安全性(可经由用户空间应用存取节点)、远程能力(可从系统中的其它处理器存取节点)以及可存取性(资源可支持多个同时客户端)。框架管理器440还可界定允许资源超驰默认框架行为(例如,请求评估或记录策略)的属性。

[0143] 随后,在框820中,框架管理器440可接收用于正被创建的特定节点601的自定义用户数据。所述用户数据可包括“C”编程语言领域中的技术人员所理解的无效“星形”字段。所属领域的技术人员还知晓用户数据为“信任我”字段。示范性自定义用户数据可包含(但不限于)例如频率表、寄存器图等表。框820中所接收的用户数据未被系统500参考,但允许在对资源的自定义未被框架管理器440辨识或完全支持的情况下进行所述自定义。此用户数据结构是“C”编程语言中既定对特定或具体使用进行扩展的基础类。

[0144] 所属领域的技术人员应认识到,用于扩展特定类的具体使用的其它种类的数据结构在本发明的范围内。举例来说,在编程语言“C++”(C加加)中,等效结构可包括将变为节点601内的资源的扩展机制的关键字“公开”

[0145] 接下来,在框825中,框架管理器440可接收相依性阵列数据。所述相依性阵列数据可包括正被创建的节点601相依的一个或一个以上资源601的唯一和特定名称。举例来说,如果正创建图6B的第一节点602,那么在此框825中,相依性阵列数据可包括第一节点602相依的第二节点622的三个资源的资源名称以及第三节点642的单个资源名称。

[0146] 随后,在框830中,框架管理器440可接收资源阵列数据。资源阵列数据可包括正被创建的当前节点的参数,例如与图6的第一节点602相关的参数(如果此第一节点602正被创建)。资源阵列数据可包括以下数据中的一者或一者以上:其它资源的名称;单位;最大值;资源属性;插入数据;以及类似于框820的自定义用户数据的任何自定义资源数据。插入数据一般识别从软件库检索到的功能,且通常列举可由正被创建的特定节点或多个节点支持

的客户端类型。所述插入数据还允许自定义客户端创建和摧毁。在框830之后,过程返回到图7的框710。

[0147] 在图8中,已用虚线说明了属性数据框815、自定义用户数据框820以及相依性阵列数据框825以指示这些特定步骤是任选的且不是任何给定节点601都需要的。同时,已用实线说明了唯一名称框805、驱动器功能框810以及资源阵列数据框830以指示例程705的这些步骤对于创建节点601是强制性的。

[0148] 图9是说明用于在PCD 100的软件架构中创建节点的图7的子方法或例程725的流程图。例程框905是根据一个示范性实施例的用于例示或创建节点601的子方法或例程725中的第一例程。在例程框905中,与正被例示的节点601相关联的一个或一个以上客户端648在此步骤中被创建。下文将结合图12进一步详细地描述关于例程框905的进一步细节。

[0149] 在框910中,框架管理器可创建或例示对应于框705的节点结构数据的一个或一个以上资源。接下来,在框915中,框架管理器440可使用在例程框705的资源阵列数据框830中所接收的最大值来激活在例程框705的例程框810中所接收的驱动器功能。根据一个示范性方面,可使用在例程框705的资源阵列数据框830中所接收的最大值来激活所述驱动器功能。根据另一优选的示范性方面,可使用连同来自例程705的节点结构数据一起被传递的任选的初始值来激活每一驱动器功能。如果未提供初始数据,那么在0(最小值)处初始化驱动器功能。还通常以知晓其正被初始化的方式来激活驱动器功能。此使得资源能够执行初始化专有的任何操作,但不需要在正常或例程操作期间执行。过程随后返回到图7的步骤730。

[0150] 图10是可由PCD 100的软件架构维持的示范性名称表1000的数据结构的图。示范性名称表1000可包括两个数据列。第一列可包括对应于资源601的别名1005且第二列可包括由框架管理器440维持以用于管理图6中所说明的节点601之间的关系的节点601的实际资源名称1010。可将多个别名用于相同资源601。别名可被感知为相依性且可在客户端的创建期间创建,如下文结合图12所描述。

[0151] 名称表1000允许集中于某些硬件和/或软件元件的第一设计小组(例如,原始设备制造商(OEM))提供相对于所述第一设计小组在内部的对硬件或软件的特定块工作的唯一名称。通过名称表1000,第二和第三(或更多)外部设计小组可能够通过使用所述第二和第三外部设计小组的硬件或软件元件所偏好的别名来参考(在此实例中的OEM的)第一设计小组的硬件或软件元件。

[0152] 举例来说,OEM可将名称“/cpu 0”指派给图5的中央处理单元402,如图10中的表1000的第一行和第二列中所说明。同时,相对于OEM的第二专业小组可能需要将不同的名称或别名指派给相同的中央处理单元402。第二小组可指派对应于资源名称“/cpu0”的“主处理器”的别名,如图10的表1000的第一行和第一列中所说明。

[0153] 图11是说明用于在PCD 100的软件架构中创建资源的别名的方法1100的流程图。框1105是用于创建资源601的别名1005的方法1100中的第一步骤。在框1105中,框架管理器440接收对应于资源601的特定名称1010的由用户选择的别名1005。接下来,在决策框1110中,框架管理器440确定由选定别名参考的资源是否已由框架管理器440创建。所属领域的技术人员还将认识到,可相对于资源或相对于另一别名来界定别名。被公布的任何名称可被别名化,而不仅仅是资源名称。

[0154] 如果对决策框1110的询问是否定的,那么遵循“否”分支到框1115,其中将别名存

储在临时存储装置中,直到创建资源为止。具体来说,当创建未界定的名称的别名时,将此别名存储在存储器中,且过程回到等待界定更多别名。当例示别名时,将别名名称连同相对于到那时未界定的名称(别名)的回调一起存储在存储器中。当公布未界定的名称时,通知别名。当存在缺少的相依性时,此行为实质上与资源创建过程相同。

[0155] 所述过程随后前进回到框1105。如果对决策框1110的询问是肯定的,那么遵循“是”分支到框1120,其中框架管理器440公布别名,使得其它资源可存取对应于刚刚已被创建的别名的资源。过程随后返回。

[0156] 图12是说明用于在PCD 100的软件架构中创建客户端648的图9的子方法或例程905的流程图。框1205是例程框905的第一步,其中创建一个或一个以上资源601的客户端648。在框1205中,框架管理器440接收指派给正被创建的客户端648的名称。类似于资源名称,客户端648的名称可包括任何类型的字母数字和/或符号。

[0157] 接下来,在框1210中,如果存在用于正被创建的此客户端648的任何特定自定义,那么框架管理器440可接收自定义用户数据。已用虚线来说明框1210以指示所述步骤是任选的。框1210的自定义用户数据类似于上文结合节点601的资源的创建所论述的自定义用户数据。

[0158] 在框1215中,框架管理器440接收指派给正被创建的特定客户端的客户端类型类别。本文的客户端类型类别可包括四种类型中的一者:(a)所需的,(b)推动,(c)向量,以及(d)等时。可取决于正由系统500管理的资源且取决于正依赖于节点601的资源的应用程序来扩展客户端类型类别。

[0159] 所需的类别一般与从所需的客户端648传递到特定资源601的标量值的处理相对应。举例来说,所需的请求可包括某一数目的每秒百万指令(“MIP”)。同时,推动类别一般与请求的处理相对应以在一定时间周期内完成某一活动而不需要开始时间或停止时间的任何指定。

[0160] 等时类别一般与对通常再现且具有良好界定的开始时间和良好界定的结束时间的动作的请求相对应。向量类别一般与通常作为需要串行或并行的多个动作的部分的数据阵列相对应。

[0161] 随后,在框1220中,框架管理器440接收指示客户端648是否已被指定为同步或异步的数据。同步客户端648是这样的客户端,即通常需要框架管理器442锁定节点601的资源,直到资源601返回资源601已结束完成来自同步客户端648的所请求任务的数据和指示为止。

[0162] 另一方面,异步客户端648可通过由框架管理器440存取的一个或一个以上线程436(参看图4)并行地处置。框架440可创建对线程436的回调,且在所述回调已由相应线程436执行时返回一值。所属领域的技术人员应认识到,异步客户端648不锁定资源,像同步客户端648在正执行同步客户端648的任务时所做的那样。

[0163] 在框1220之后,在决策框1225中,框架管理器440确定由客户端645识别的资源是否可用。如果对决策框1225的询问是否定的,那么遵循“否”分支到框1230,其中将空值或消息返回到用户,从而指示无法在此时创建客户端648。

[0164] 如果对决策框1225的询问是肯定的,那么遵循“是”分支到决策框1235,其中框架管理器440确定由客户端648识别的每一资源是否支持框1210中所提供的客户端类型。如

果对决策框1235的询问是否定的,那么遵循“否”分支回到框1230,其中返回空值或消息,从而指示无法在此时创建客户端648。

[0165] 如果对决策框1235的询问是肯定的,那么遵循“是”分支到框1240,其中框架管理器440在存储器中创建或例示客户端648。接下来,在框1245中,如果在框1210中接收到任何自定义用户数据,例如任选的自变量,那么可用其相应的资源将这些任选的自变量映射到特定节点601。接下来,在框1250中,新创建的客户端645耦合到处于闲置状态中或在所请求的状态上的其对应的一个或一个以上资源,如上文所描述的图6B中所说明。过程随后返回到图9的框910。

[0166] 图13是说明用于在PCD 100的软件架构中创建针对资源601的客户端请求675的方法1300的流程图。方法1300一般在如上文结合图7和图12所描述的客户端创建和节点创建之后执行。框1305是用于创建针对资源601的客户端请求675的方法1300中的第一步骤。此方法1300将描述如何通过框架管理器440来处置以下三种类型的请求675:(a)所需的,(b)推动以及(c)向量。下文将结合图15来描述对第四种类型的请求675((d)等时请求)的处置。如上文所提及的请求675的名称所暗示,客户端请求675一般与上文结合图12所创建和描述的客户端类型相对应。

[0167] 在框1305中,框架管理器440可接收与特定客户端请求675相关联的数据,所述客户端请求例如为上文提及的三种请求中的一者:(a)所需的,(b)推动以及(c)向量。与所需的请求相关联的数据一般包括从所需的客户端648传递到特定资源601的标量值。举例来说,所需的请求可包括某一数目的每秒百万指令(“MIP”)。同时,推动包括用以在一定时间周期内完成某一活动而不需要开始时间或停止时间的任何指定的请求。用于向量请求的数据一般包括需要串行或并行地完成的多个动作的阵列。向量请求可包括任意长度的值。向量请求通常具有大小值和值阵列。节点601的每一资源可经扩展以具有指针字段,以便支持向量请求。在“C”编程语言中,所述指针字段由所属领域的技术人员所理解的联合功能支持。

[0168] 客户端请求数据还可指示所述请求是否为可压制的。可压制请求可由资源根据资源的自身选择来不同地处置,如上文结合图7D的方法700D所描述。

[0169] 接下来,在框1310中,框架管理器440通过由上文结合图13所描述的方法创建的客户端648来发出请求。随后,在框1315中,如果所述请求是所需的类型或向量类型,那么框架管理器440加倍缓冲正穿过客户端的所述请求数据。如果所述请求是推动类型,那么框架管理器440跳过框1315。

[0170] 对于所需的请求,在此框1315中,来自前一请求的值被维持在存储器中,使得框架管理器440可确定在先前所请求值与当前组的所请求值之间是否存在任何差异。对于向量请求,通常不将先前请求维持在存储器中,但节点601的资源可针对特定实施方案在需要时维持所述先前请求。因此,框1315对于向量类型的请求是任选的。

[0171] 在框1320中,框架管理器440计算所请求值中的先前组与所请求值的当前组之间的增量或差。在决策框1325中,框架管理器确定当前组的所请求值是否等同于先前组的所请求值。换句话说,框架管理器440确定当前组的所请求值与先前组的所请求值之间是否存在差。如果在当前组的所请求值与先前组的所请求值之间不存在差,那么遵循“是”分支(其跳过框1330到框1370)到框1375,其中过程结束。

[0172] 如果对决策框1325的询问是否定的,意味着所述组的所请求值相对于所述组的先前所请求值是不同的,那么遵循“否”分支到决策框1330。

[0173] 在决策框1330中,框架管理器440确定读取请求是否为异步请求。如果对决策框1330的询问是否定的,那么遵循“否”分支到框1340,其中框架管理器440锁定对应于客户端请求675的资源601。如果对决策框1330的询问是肯定的,意味着读取请求是异步请求类型,那么遵循“是”分支到框1335,其中可将所述请求推到另一线程上且在框架管理器440当前管理多核系统(比如图4的多核系统)的情况下可由另一核执行。已用虚线说明框1335以指示如果PCD 100是单核中央处理系统,那么此步骤可为任选的。

[0174] 随后,在框1340中,框架管理器440锁定对应于请求675的资源601。接下来,在框1345中,资源601执行一般对应于在图8的框830中所接收的资源阵列数据的插入数据的更新功能所述更新功能一般包括按照新客户端请求来负责新资源状态的功能。所述更新功能将其先前状态与客户端请求中的所请求状态进行比较。如果所请求的状态大于先前状态,那么更新功能将执行客户端请求。然而,如果所请求的状态等于或小于当前状态以及资源正操作的状态,那么客户端请求将不被执行以便增加效率,因为老的状态实现或满足所请求的状态。更新功能从客户端取得新的请求且将所述请求与所有其它有效请求聚集在一起以确定资源的新状态。

[0175] 作为一实例,多个客户端可请求总线时钟频率。所述总线时钟的更新功能将通常采用所有客户端请求中的最大者且将其用作总线时钟的新的所需状态。这不是所有资源将使用相同更新功能的情况,但存在将由多个资源使用的一些更新功能。一些共同更新功能将采用客户端请求中的最大者、采用客户端请求中的最小者,且对客户端请求求和。或者资源可界定其自身的自定义更新功能(如果其资源需要以某一独特的方式聚集请求)。

[0176] 接下来,在框1350中,框架管理器440将数据传递到对应于客户端请求648的资源,使得所述资源可执行节点601的资源专有的驱动器功能。驱动器功能应用更新功能所计算的资源状态。此可能需要更新硬件设定、向相依资源发出请求、调用传统功能,或以上各者的某一组合。

[0177] 在先前实例中,更新功能计算所请求的总线时钟频率。驱动器功能可接收所述所请求的频率且其可更新所述时钟频率控制HW在所述频率下允许。应注意,有时驱动器功能可能不满足更新功能已计算的确切所请求状态。在此情况下,驱动器功能可选择最佳满足请求的频率。举例来说,总线时钟HW可仅能够在128MHz和160MHz下运行,但所请求的状态可为150MHz。在此情况下,驱动器功能应160MHz下运行,因为其超过所请求的状态。

[0178] 接下来,在框1355中,框架440从已在框1350中执行驱动器功能的资源接收状态控制。随后,在框1360中,如果针对资源进行界定,那么事件690可经触发,使得将数据传递回对应于事件690的客户端648。可在另一线程中处理事件。此可使锁定的资源所花费的时间量最少,且允许如图4中所说明的多核系统中的更并行的操作。可以类似于如此方法1300中所描述的如何针对资源来界定请求的方式来针对资源界定一个或一个以上事件690。换句话说,事件创建过程可使客户端创建过程最大程度地并行。与事件不同的一件事是有可能界定仅在越过某些阈值时才会被触发的事件。

[0179] 仅基于阈值才会被触发的事件的此界定实现以下通知:在资源变得超预订时(其具有比其可支持的同时用户多的同时用户),其指示系统过载条件;或当资源变低/没有时,



其可允许其它物被关闭,恢复在系统变得超预订时所停用的功能性等。因为可用阈值完成事件登记,所以其减少系统必须对事件通知所做的工作量,而仅在存在一些确实必需的事情时才发生。还有可能针对每个状态改变的事件进行登记。

[0180] 接下来,在任选的框1365中,如果正被处理的请求是向量请求,那么通常执行此任选的框1365。任选的框1365一般包括检查或确定以评估向量指针是否仍定位在用户传递到向量中的相同数据上。如果对此任选的框1365的询问是肯定的,意味着指针仍指向用户传递到向量中的相同数据,那么清除指针,使得对旧数据的参考不被维持。一般执行此任选的框1365来虑及在上文在正处理向量请求时所描述的与推动请求和所需的请求进行比较的加倍缓冲框1315。

[0181] 随后,在框1370中,框架440解开所请求的资源,使得其它客户端请求648可被特定节点601的当前的但现在被释放的所请求资源处置。所述过程随后返回到第一框1305以用于接收下一客户端请求。

[0182] 图14是说明对PCD 100的资源的等时客户端请求中所请求的工作的图1400。图1400包括具有X轴和Y轴的图表。X轴一般包括在Y轴可包括所请求的动作值(例如,所请求的每秒百万指令(MIP))时所逝去的事件。如先前所述,等时客户端请求一般包括良好界定的开始时间A和良好界定的结束时间或最终期限C。在图14中所说明的示范性实施例中,图表指示150MIP的所请求工作在时间A处开始且在时间B处结束,其中时间B在时间C的所请求的最终期限之前出现。

[0183] 图15是说明用于在PCD 100的软件架构中创建针对资源的等时客户端请求的图9的子方法或例程1300B的流程图。此子方法或例程1300B是从上文结合图13所描述的步骤建立或与所述步骤结合执行。这意味着此图15中所列举的步骤以对应于图13中所提供的参考数字的序列定位。如下文更详细地描述,在步骤的次序或序列不影响来自所执行的步骤的所需输出时,本发明不限于此次序或序列。

[0184] 框1307是用于处理等时请求675的子方法或例程的第一步骤。框1307在框1305之后且在图13的框1310之前出现。在框1307中,框架管理器440可接收最终期限数据,例如上文结合图14所论述的最终期限C。

[0185] 接下来,在框1309中,框架管理器440可计算当前时间与在框1307中提供的最终期限之间的差。随后在发生在图13的框1360之后但在框1365之前的框1362中,框架管理器440将开始时间A和结束时间B与最终期限C进行比较(参看图14)。在框1363中,因为框架管理器440具备所请求的活动量且因为框架管理器440跟踪开始时间A和结束时间B,于是在框1363中,框架管理器440可计算由特定节点601的资源执行的工作量。

[0186] 接下来,在发生在图13的框1365之后且在框1370之前的框1367中,可执行优化过程。已用虚线说明框1367以指示所述步骤是任选的,或此步骤可离线且离开与PCD100相关的装置执行。所述优化过程可试图确定如何在最终期限中的开始时间之间最佳地完成工作,同时考虑到许多不同变量,例如电力消耗和响应性。在一些示范性实施例中,在不脱离本发明的范围的情况下,可完全彻底地跳过此框1367。所述过程随后返回到图13的框1305以用于处理下一客户端请求675。

[0187] 图16是说明用于在PCD 100的软件架构500中创建针对资源的阈值事件690的方法1600的流程图。图16一般与上文结合图6C、6D所描述的阈值事件690相对应。如先前所述,虽

然图6C中所说明的阈值事件690与温度相关,但所属领域的技术人员应认识到,阈值事件690可与可在PCD 100内监视的任何类型的条件相关联。

[0188] 举例来说,其它阈值事件包含(但不限于)与电力消耗、电池电荷状态/状况、休眠或低电力条件、特定硬件条件(例如,良好、好和差射频(“RF”)接收)相关的阈值事件、可与比如每秒百万指令(“MIP”)、时钟频率、总线频率、电压、DCVS条件和算法等负载相关的操作条件。阈值事件690通常包含一个或一个以上触发条件。在图6C所说明的实例中,所述触发条件与特定温度值相关。

[0189] 现在返回参看图16,框1605是用于创建一个或一个以上阈值事件690的方法1600的第一步骤。方法1600一般在如图7中所说明的用于管理PCD 100的资源的方法700之后实行或执行。在框1605中,图5A的框架管理器440可接收用以针对节点601的资源创建阈值事件690的一个或一个以上请求。

[0190] 如上文所述,由框架管理器440支持的软件架构500的任何用户可针对资源设置阈值事件690。创建阈值事件690的软件架构500的用户可包含资源创造者、节点创造者、资源,或执行线程。可指令指派给阈值事件690的资源向一个或一个以上不同资源报告阈值事件的状态。

[0191] 接下来,在框1610中,用于40A的框架管理器接收跟踪针对节点601的一个或一个以上资源的事件/条件的一个或一个以上功能。此框1610一般与图6C中所说明的阈值事件690相对应。在框1615中,框架管理器440将接收正被一个或一个以上功能跟踪的一个或一个以上阈值或条件。具体来说,对于框1610和1615,框架管理器440将接收图6C中所说明的阈值事件690A到D中的一者。图6C中所说明的每一阈值事件690具有温度监视功能以及正被所述功能跟踪的特定温度。

[0192] 接下来,在框1620中,框架管理器440将接收一个或一个以上资源的一个或一个以上名称,所述一个或一个以上资源将接收阈值事件的通知。框1620一般与图6D中所说明的图表697相对应。图表697在第三列中列出用于第二列中所列出的阈值事件的接收方资源。

[0193] 接下来,在框1625中,用框架管理器440登记针对资源的一个或一个以上阈值事件。随后,在框1630中,每一资源开始其对方法1600中所界定的其阈值事件的监视。方法1600随后返回。

[0194] 图17是说明用于在PCD 100的软件架构500中用新的(实际)资源取代存根资源的方法1700的流程图。图17一般与图8的其中可创建存根资源的框803相对应。如先前结合图8所述,存根资源是允许资源创造者在图6A的框架或资源图表500B1中创建占位符的软件开发工具。存根资源是可允许形成其它资源且其它资源可相依的占位符。以此方式,当特定资源不可用或尚未开发出时,可创建其它资源和最终的节点和客户端。

[0195] 节点的其它资源可登记针对存根资源的事件。然而,存根资源将从不返回事件的值,因为存根资源将不含有可支持事件的任何有效功能或特征。可用如在图17中所设定的实际或运行资源来取代存根资源。

[0196] 框1705是用于用新的(实际)起作用的资源取代存根资源的方法1700中的第一步骤。在框1705中,用于40的框架管理器可接收存根资源相依的一个或一个以上资源的身份。接下来,在框1710中,用于40的框架管理器可锁定所述存根资源相依的一个或一个以上资源,使得这些其它资源可不发出任何进一步请求。

[0197] 接下来,在框1715中,可将在用于存根资源的数据结构中列出的每一客户端添加到将取代存根资源的新的起作用资源的列表。在框1720中,调整存根资源的每一客户端的指针,使得其指向正取代存根资源的新的起作用资源。

[0198] 在框1725中,框架管理器440可发出对老的存根资源相依的每一资源的一个或一个以上有效请求。接下来,在框1730中,框架管理器440可重新聚集正取代老的存根资源的新资源的新状态。接下来,在框1735中,在框915中,框架管理器440可使用在例程框705的资源阵列数据框830中所接收的最大值来激活在例程框705的例程框810中所接收的驱动器功能。

[0199] 根据一个示范性方面,可使用在例程框705的资源阵列数据框830中所接收的最大值来激活所述驱动器功能。根据另一优选的示范性方面,可使用连同来自例程705的节点结构数据一起被传递的任选的初始值来激活每一驱动器功能。如果未提供初始数据,那么在0(最小值)处初始化驱动器功能。还通常以知晓其正被初始化的方式来激活驱动器功能。此使得资源能够执行初始化专有的任何操作,但不需要在正常或例程操作期间执行。

[0200] 接下来,在框1740中,框架管理器440可针对取代存根资源的新的以及功能性资源而重新发出先前针对存根资源而登记的一个或一个以上事件。过程1700随后返回。

[0201] 在本说明书中所描述的过程或过程流中的某些步骤自然先于供本发明如所描述而起作用的其它方面。然而,如果此类次序或序列不更改本发明的功能性,则本发明不限于所描述的步骤的次序。也就是说,应认识到,在不脱离本发明的范围和精神的情况下,一些步骤可在其它步骤之前、之后或与其它步骤并行地(大体上同时地)执行。在一些情况下,在不脱离本发明的情况下,可省略或不执行某些步骤。此外,例如“其后”、“随后”、“接下来”等词语无意限制步骤的次序。这些词语仅用于引导读者阅读对示范性方法的描述。

[0202] 另外,编程领域中的技术人员能够编写计算机代码或识别适当的硬件和/或电路以基于(例如)本说明书中的流程图和相关联的描述而无困难地实施本发明。

[0203] 因此,对于充分理解如何制作和使用本发明来说,对程序代码指令或详细硬件装置的特定集合的揭示不被视为必需的。在以上描述中且结合可说明各种过程流的图更详细地阐释所主张的计算机实施的过程的发明性功能。

[0204] 在一个或一个以上示范性方面中,可以硬件、软件、固件或其任何组合来实施所描述的功能。如果以软件实施,则可将功能作为一个或一个以上指令或代码而在计算机可读媒体上进行存储或传输。计算机可读媒体包含计算机存储媒体与包括促进计算机程序从一处传递到另一处的任何媒体的通信媒体两者。存储媒体可为可由计算机存取的任何可用媒体。举例来说而非限制,此计算机可读媒体可包含RAM、ROM、EEPROM、CD-ROM或其它光盘存储装置、磁盘存储装置或其它磁性存储装置,或可用以载运或存储呈指令或数据结构形式的所要程序代码且可通过计算机存取的任何其它媒体。

[0205] 同样,可适当地将任何连接称作计算机可读媒体。举例来说,如果使用同轴电缆、光纤电缆、双绞线、数字订户线(“DSL”)或例如红外线、无线电及微波的无线技术从网站、服务器或其它远程源传输软件,则同轴电缆、光纤电缆、双绞线、DSL或例如红外线、无线电及微波的无线技术包含于媒体的定义中。

[0206] 如本文中所使用,磁盘及光盘包括压缩光盘(“CD”)、激光光盘、光学光盘、数字多功能光盘(“DVD”)、软盘和蓝光光盘,其中磁盘通常以磁性方式再现数据,而光盘用激光以

光学方式再现数据。以上各者的组合也应包含在计算机可读媒体的范围内。

[0207] 虽然已详细地说明和描述了选定的方面,但将理解,在不脱离由所附权利要求书界定的本发明的精神和范围的情况下,可在其中进行各种替代和更改。

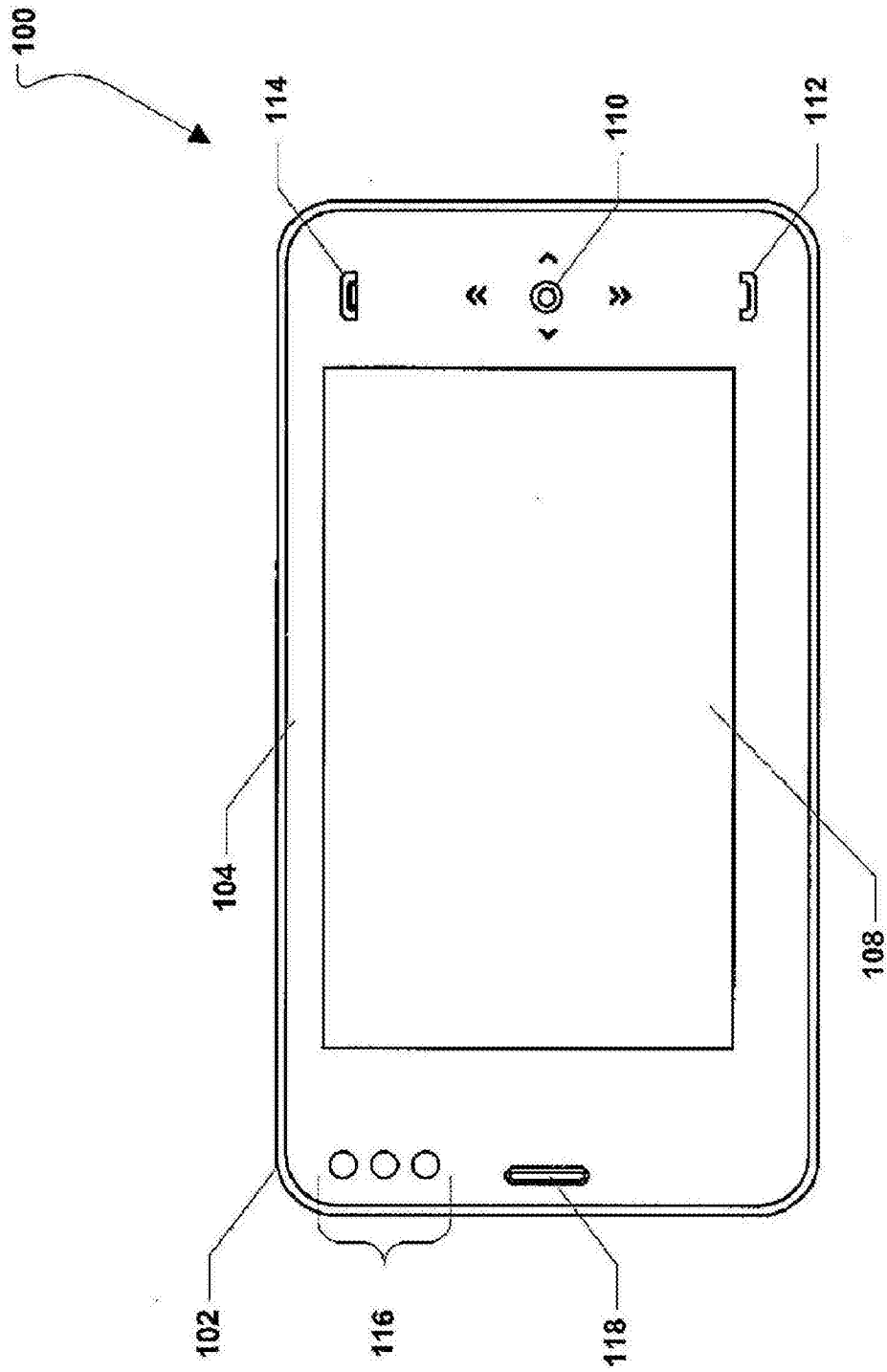


图1

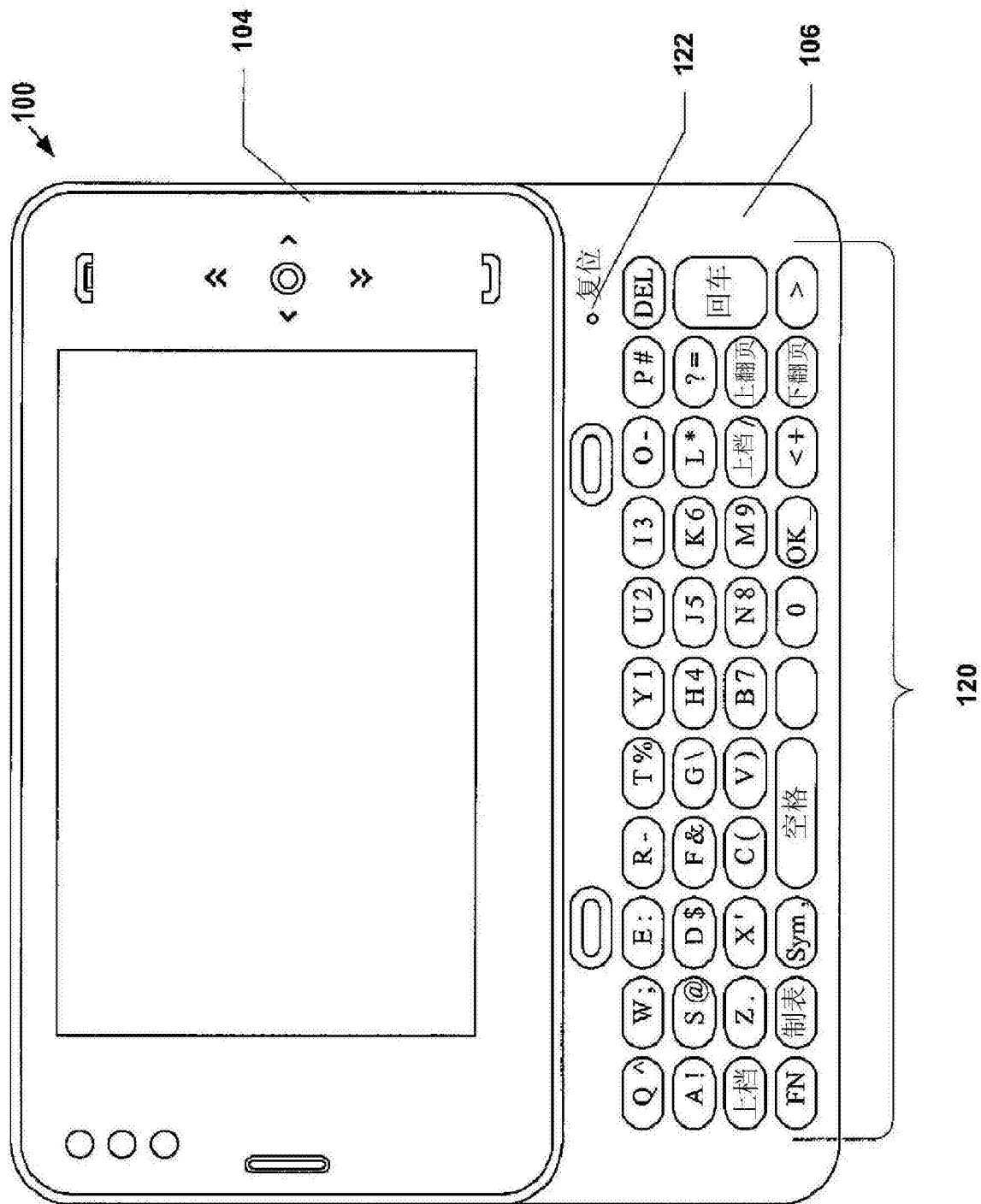


图2

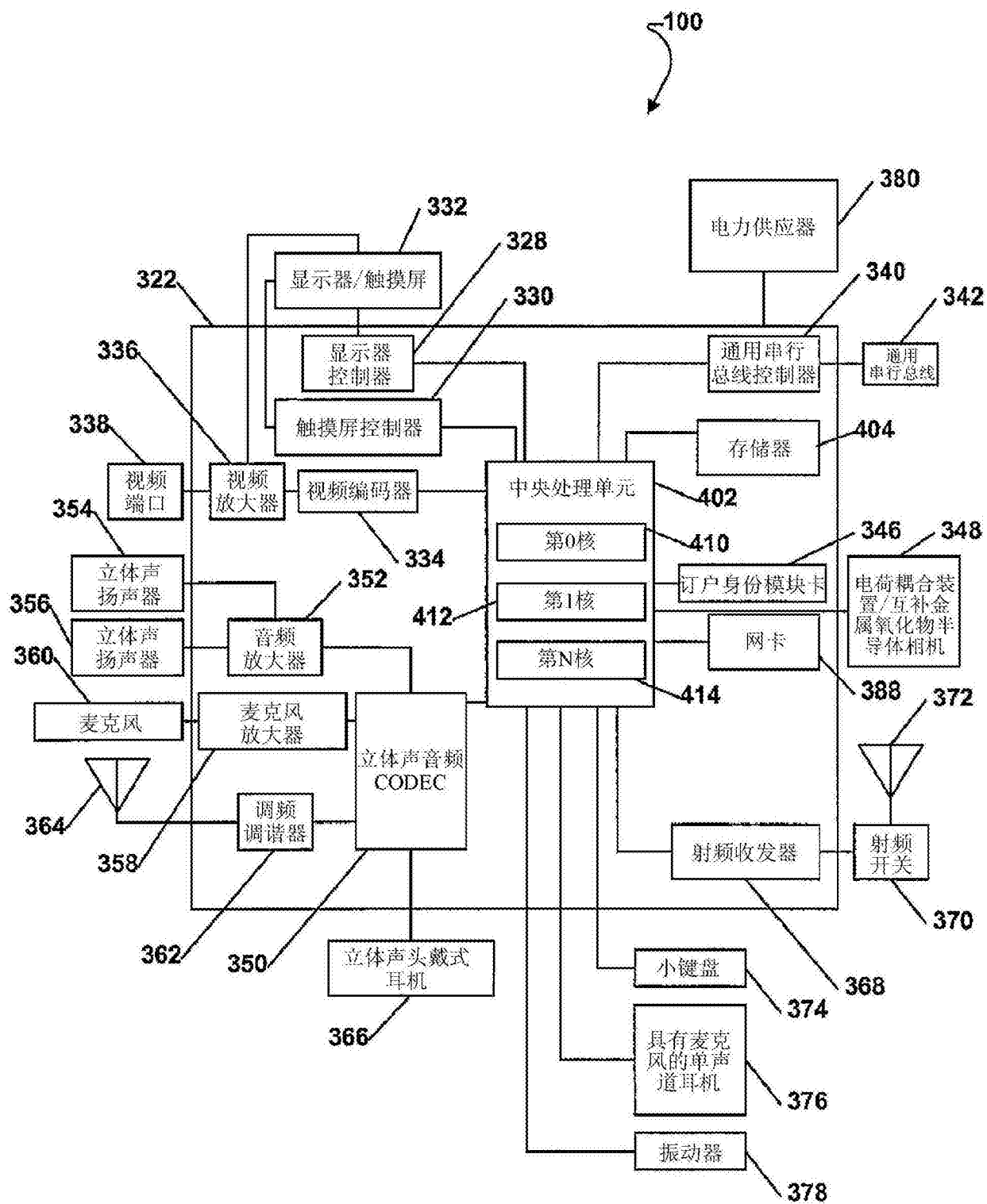


图3

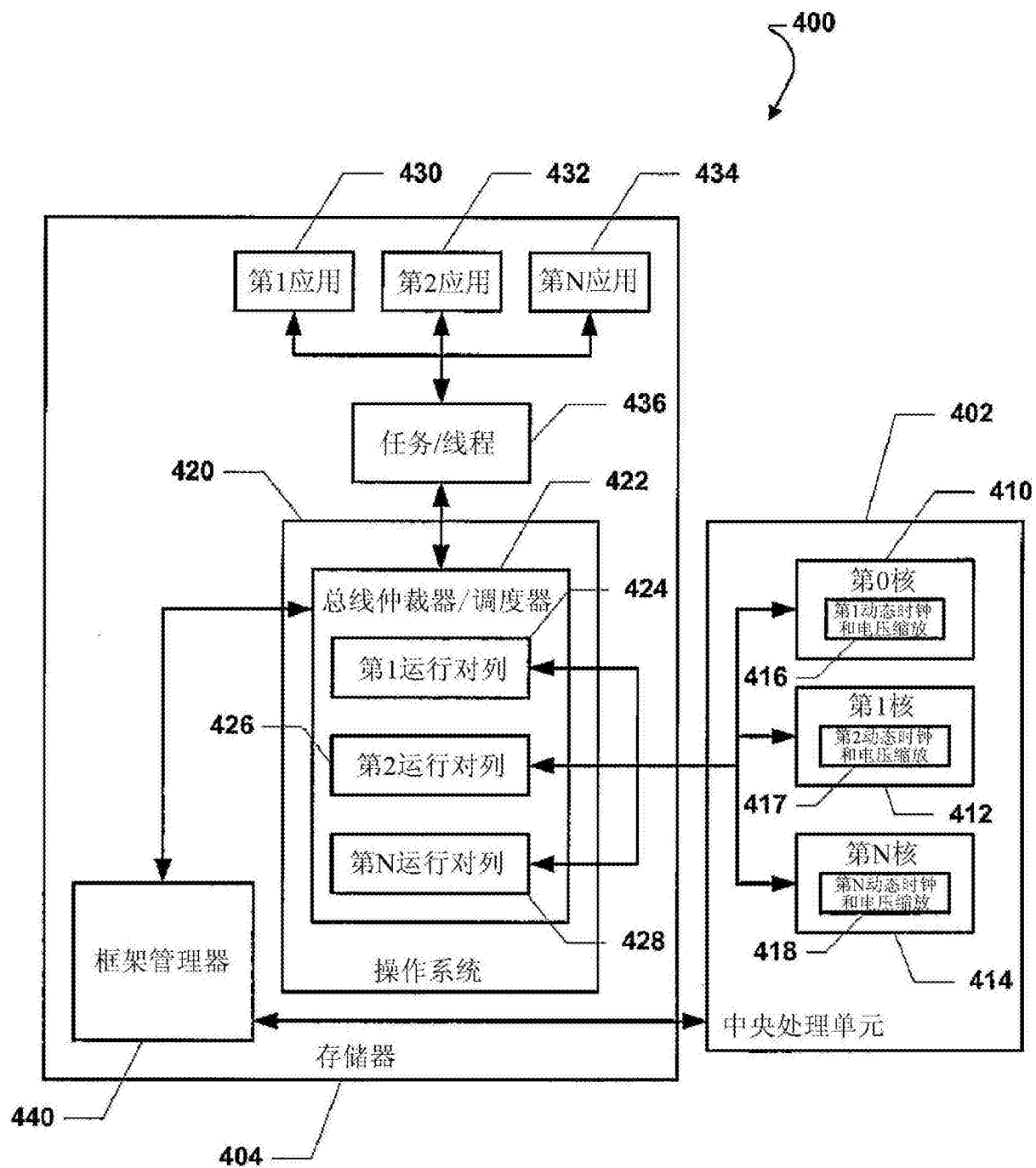


图4



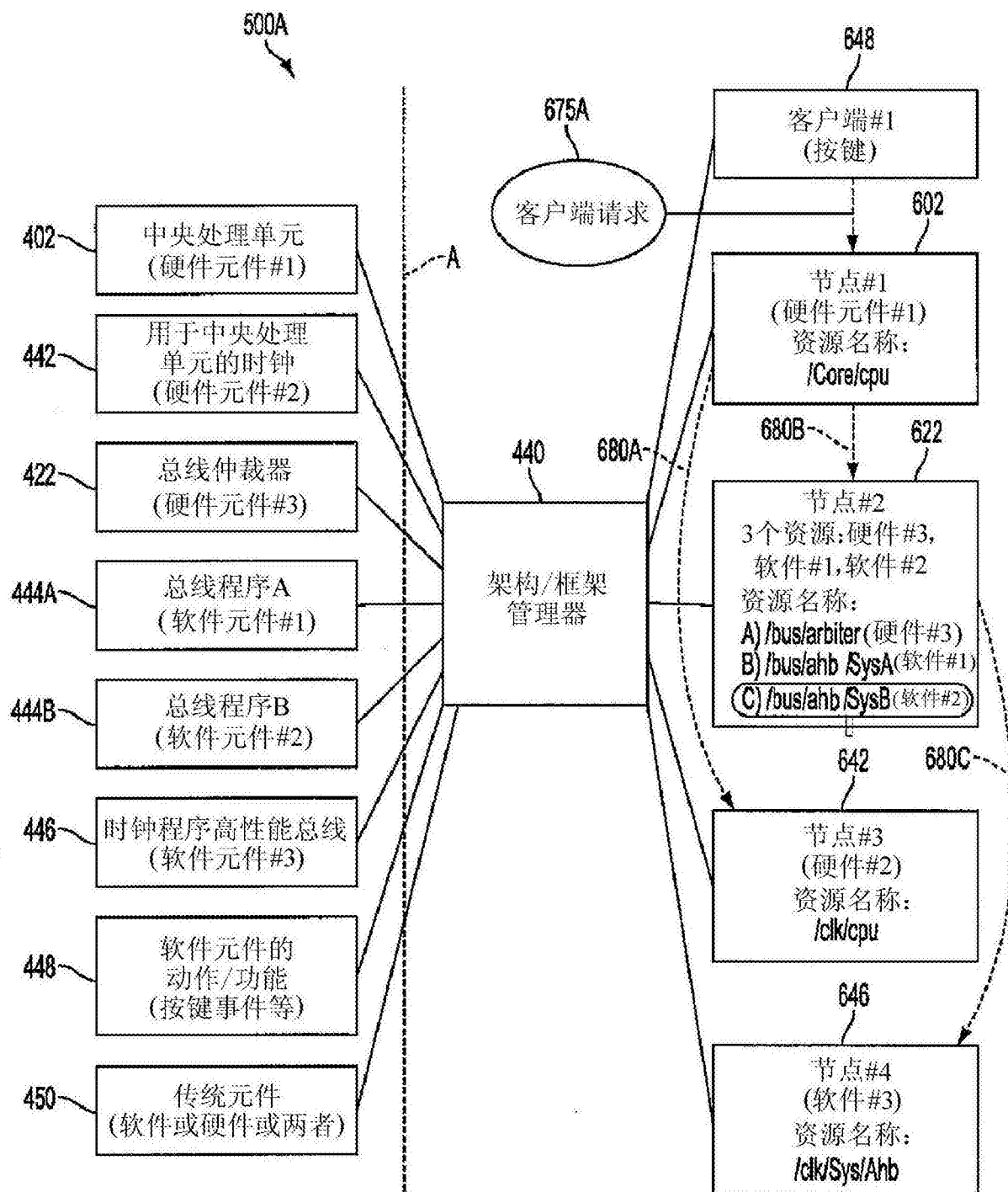


图5

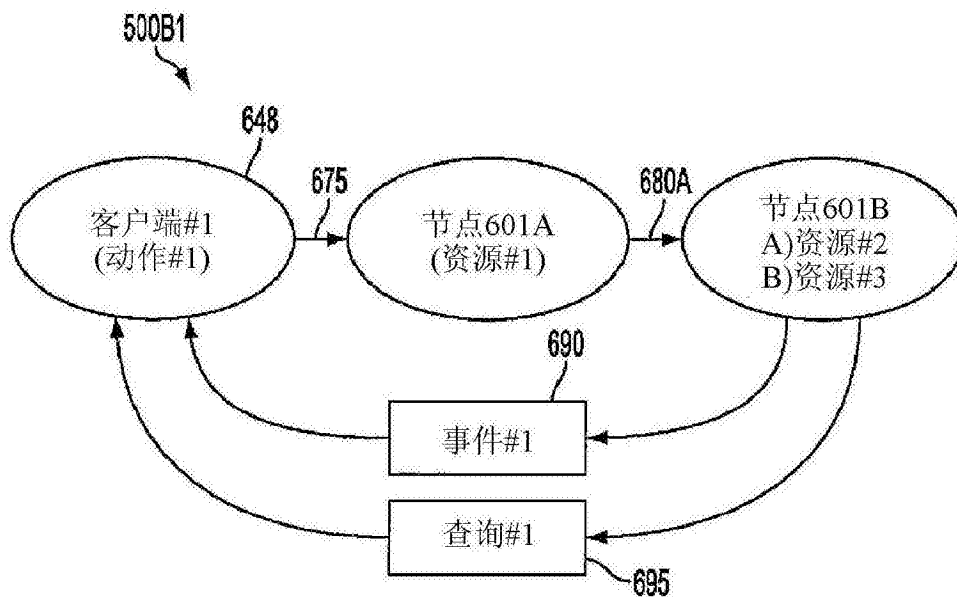


图6A

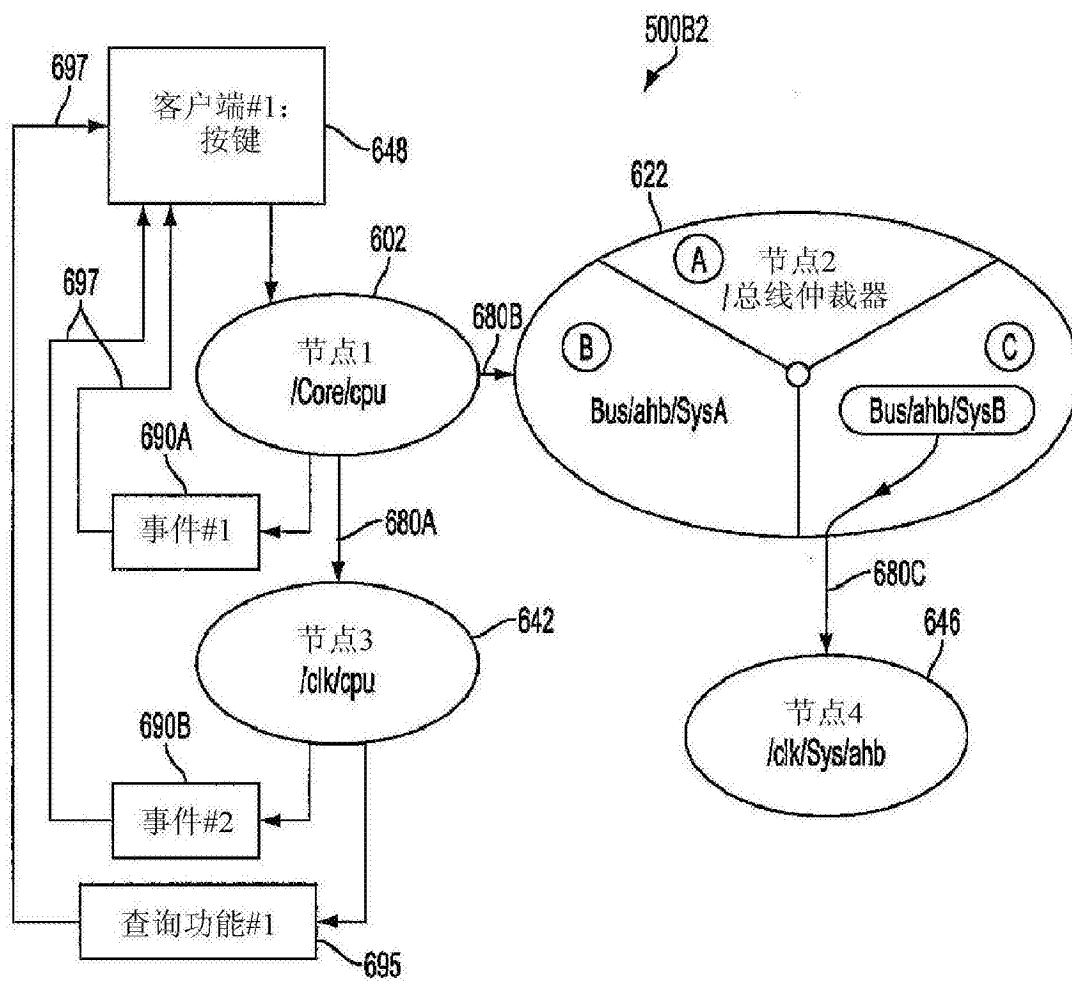


图6B

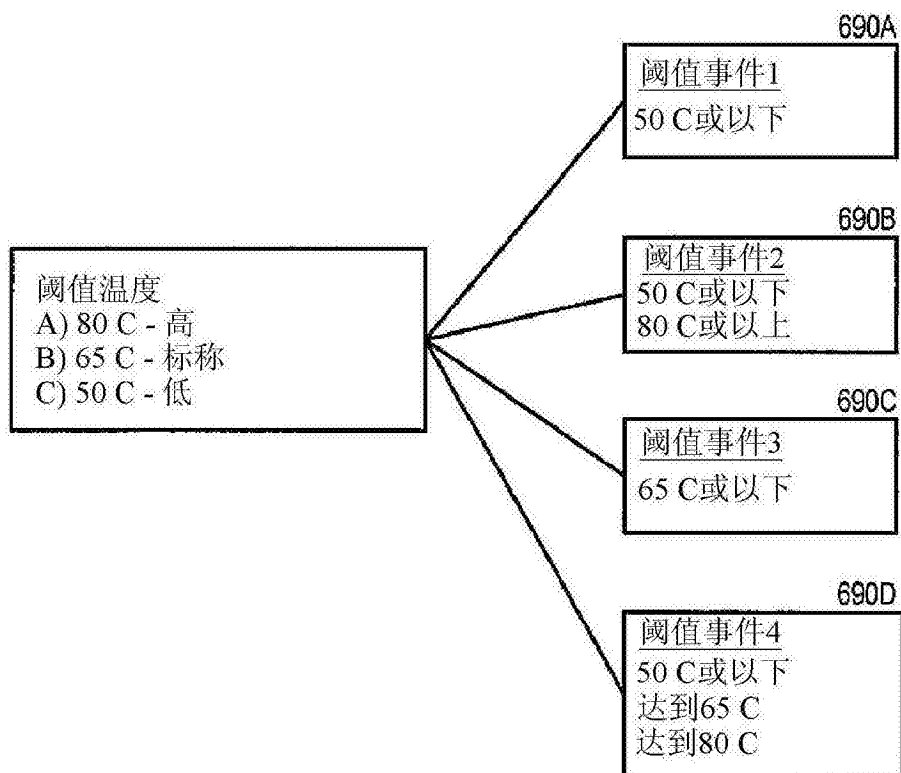


图6C

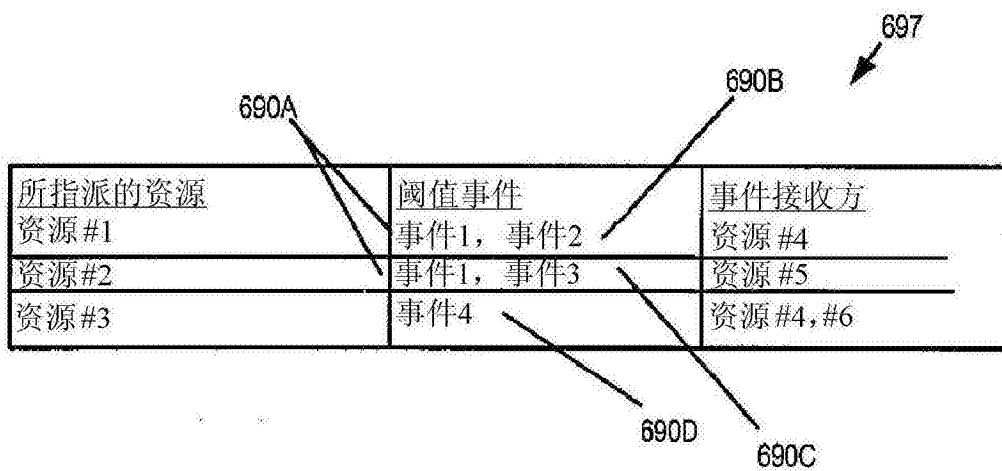


图6D

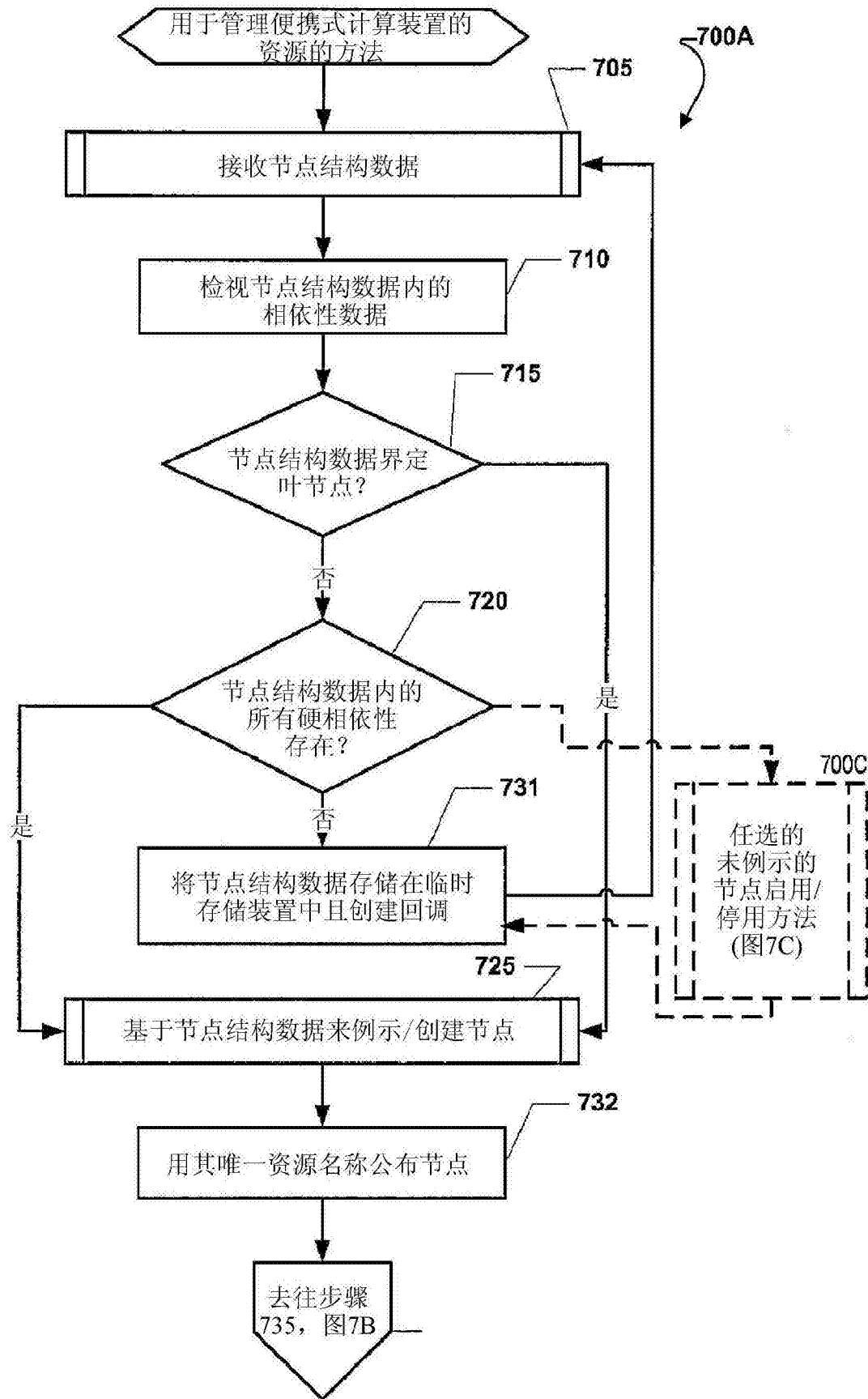
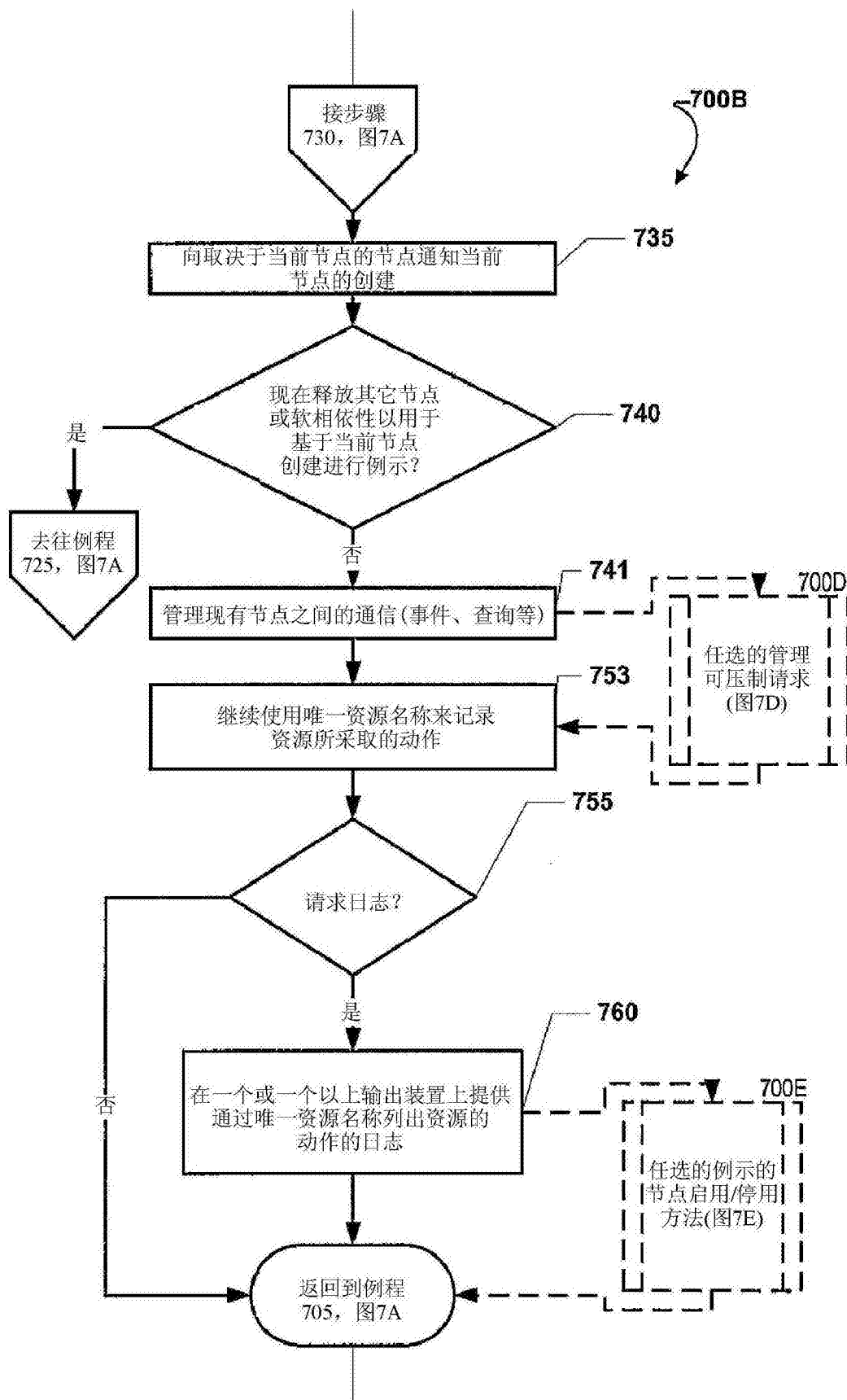


图7A



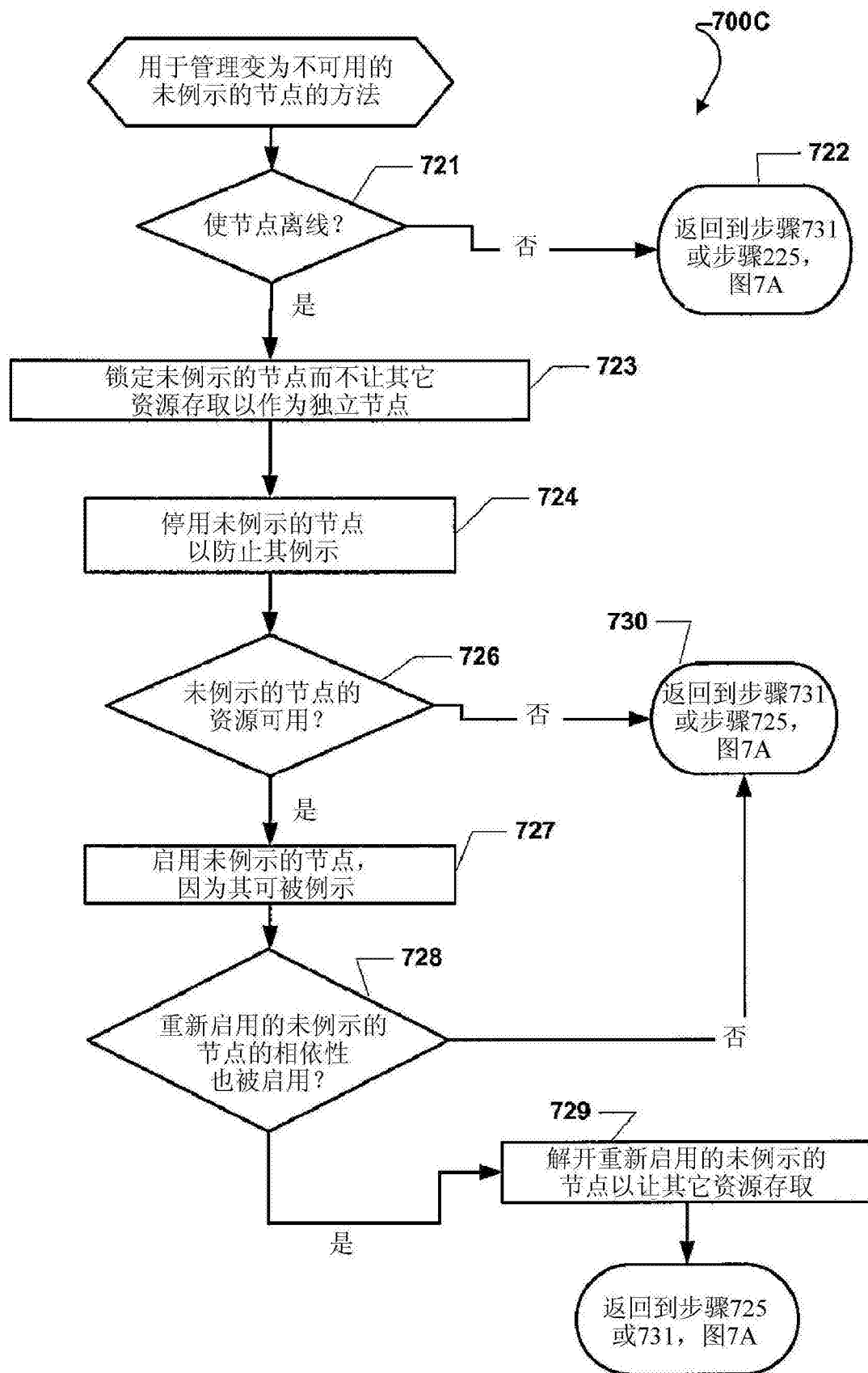


图7C

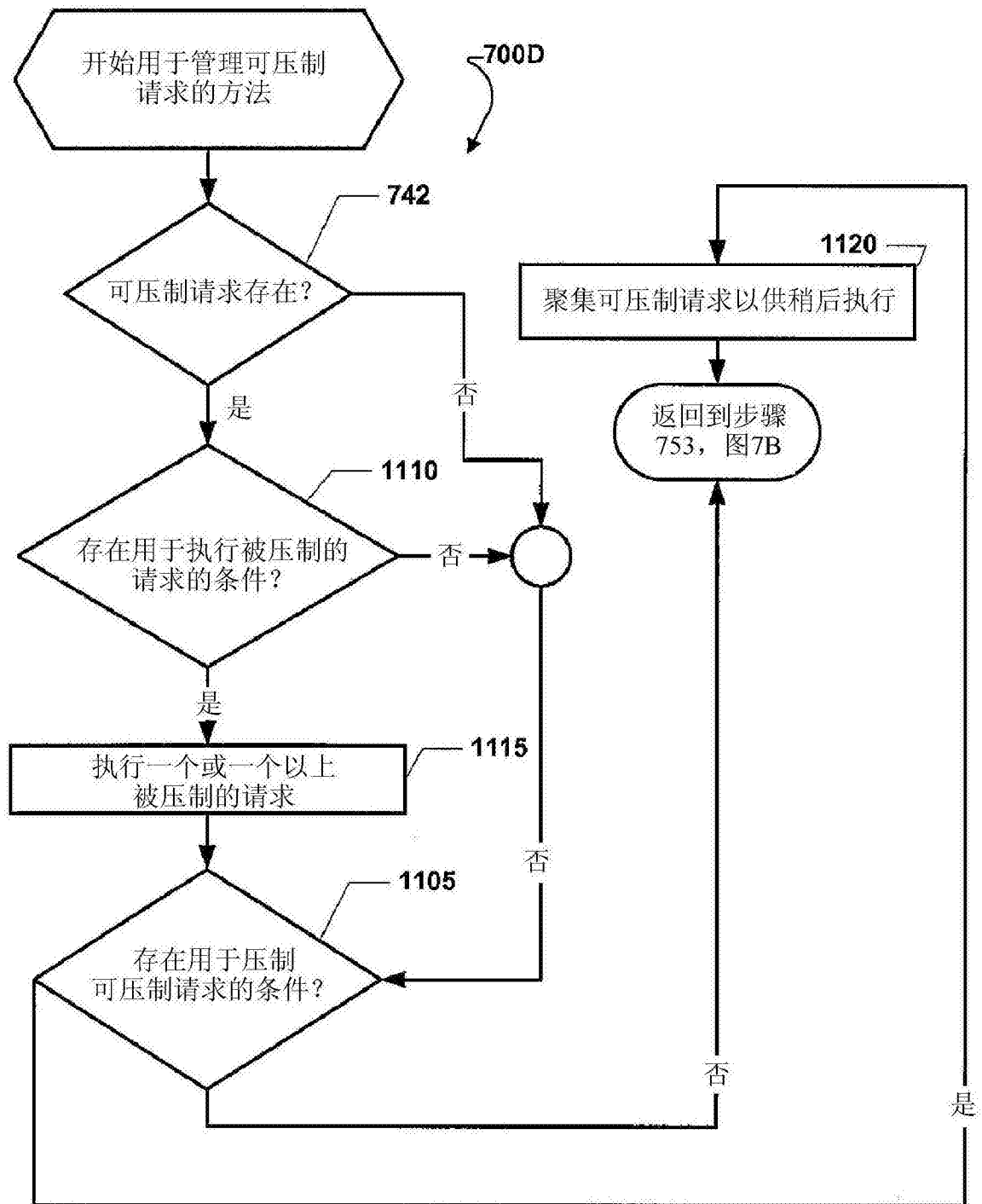


图7D

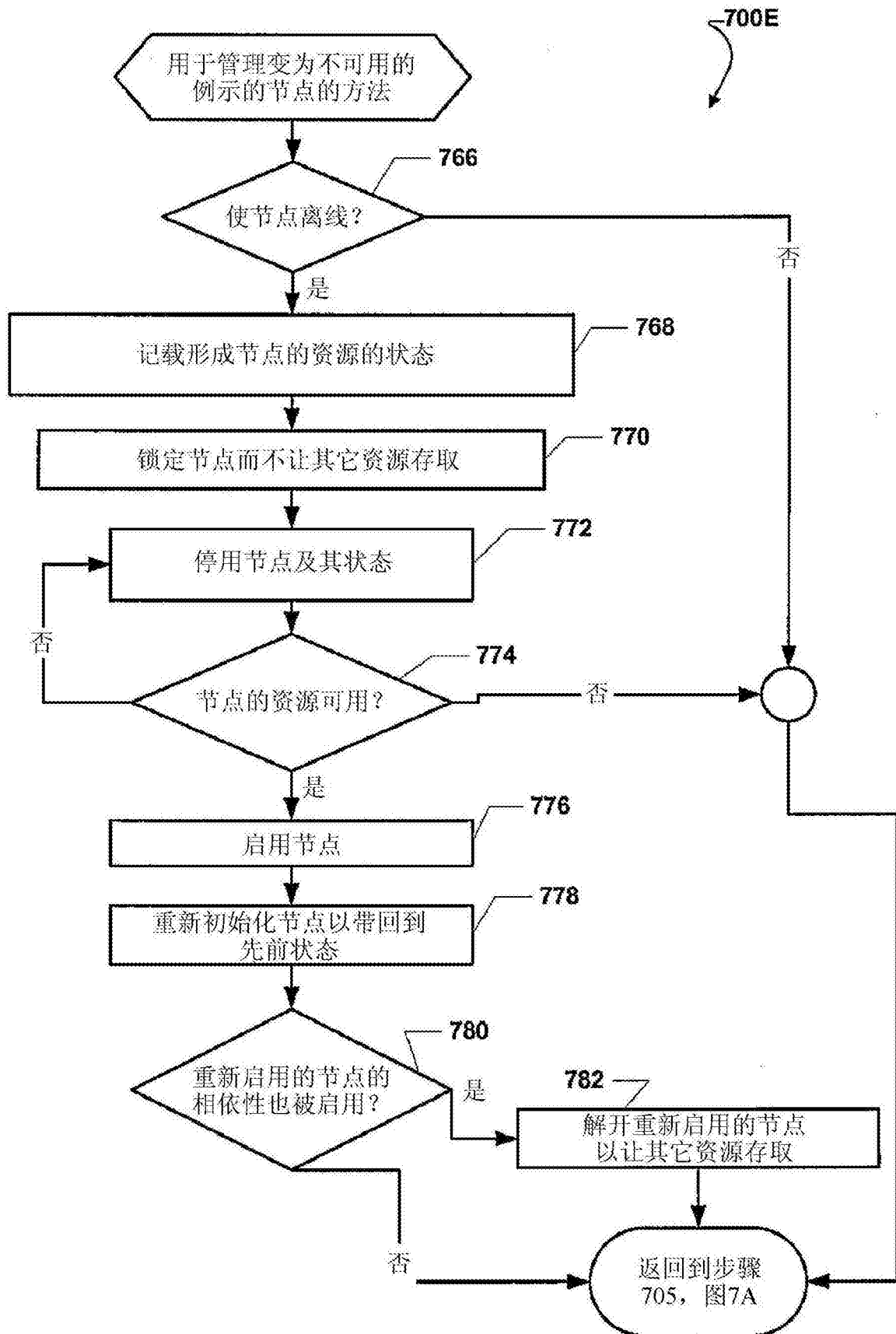


图7E



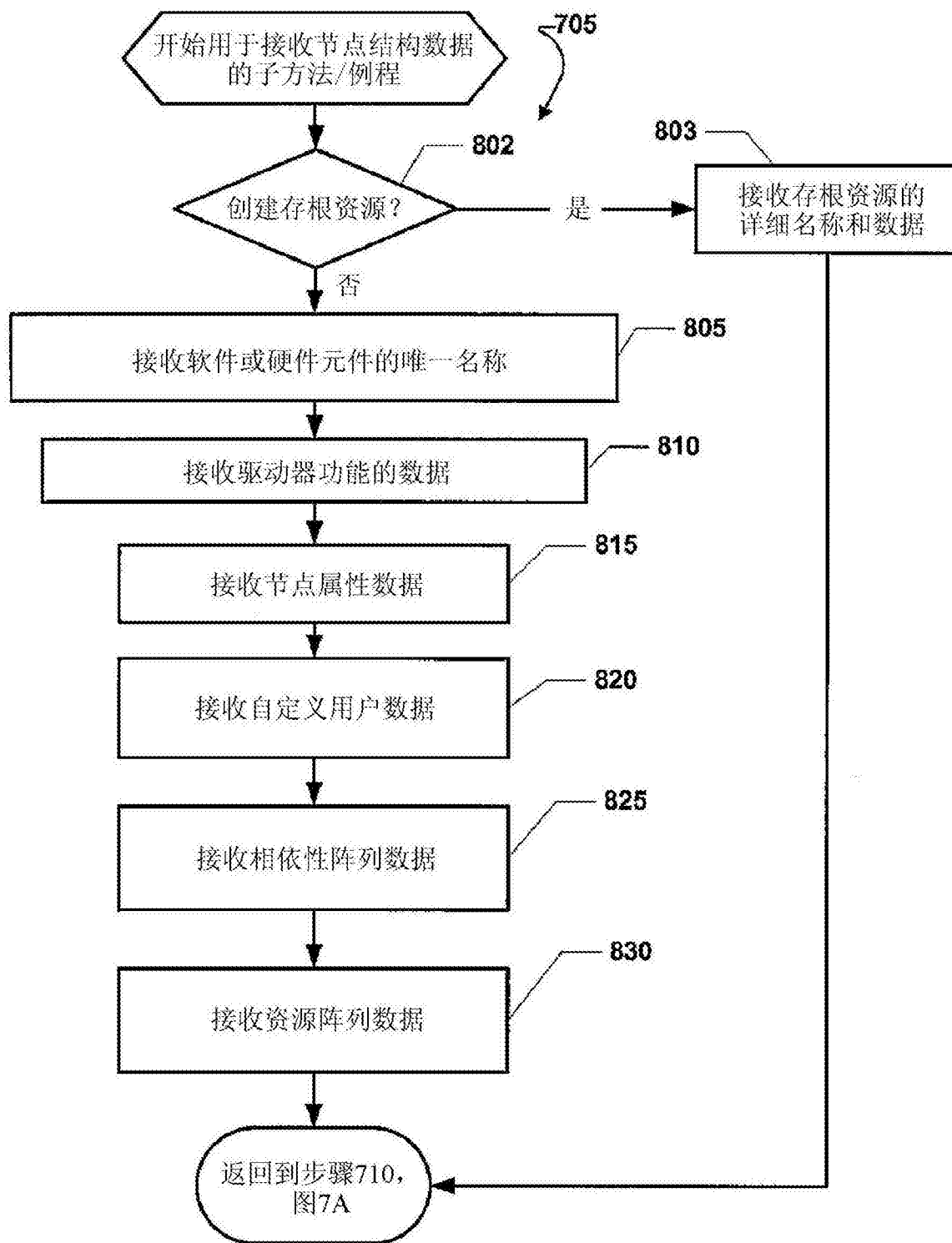
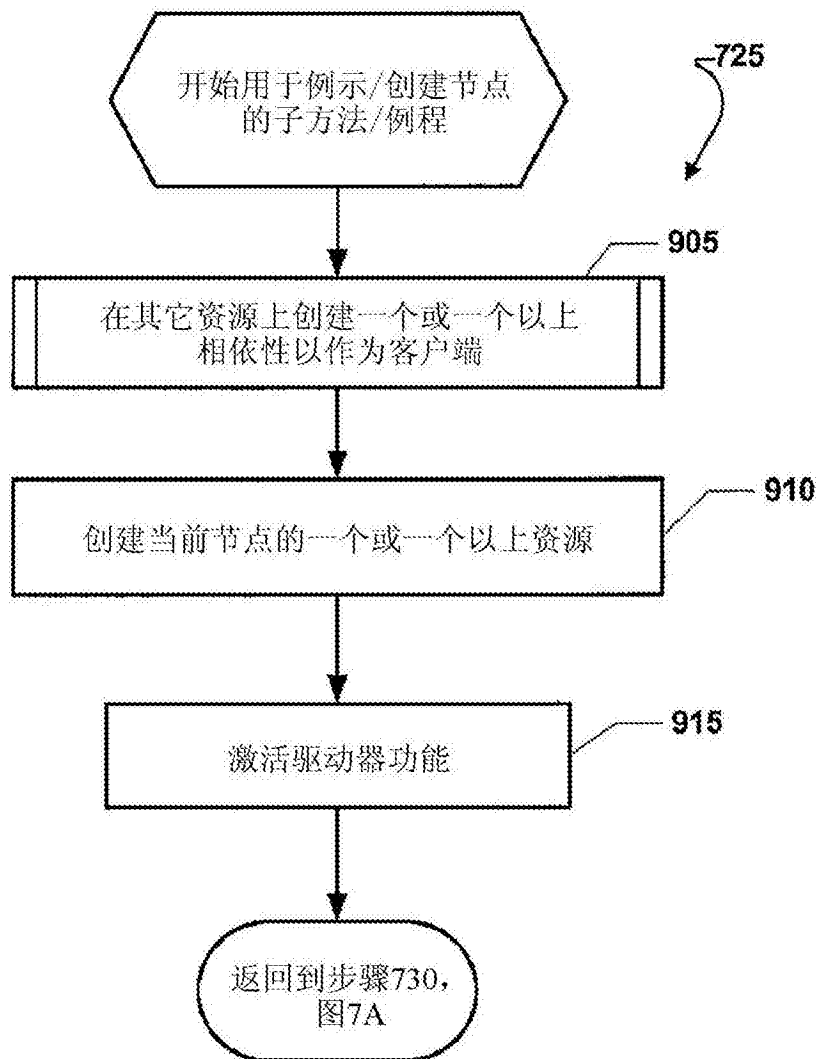


图8



名称表

| 别名        | 资源名称         |
|-----------|--------------|
| 主处理器      | 中央处理单元0(硬件)  |
| 时钟程序高性能总线 | 高性能总线(软件)    |
| 总线程序A     | /bus/ah/sysA |
| 电力/相机     | /电力轨1(硬件)    |
| 电力/中央处理单元 | /电力轨2(硬件)    |

图10

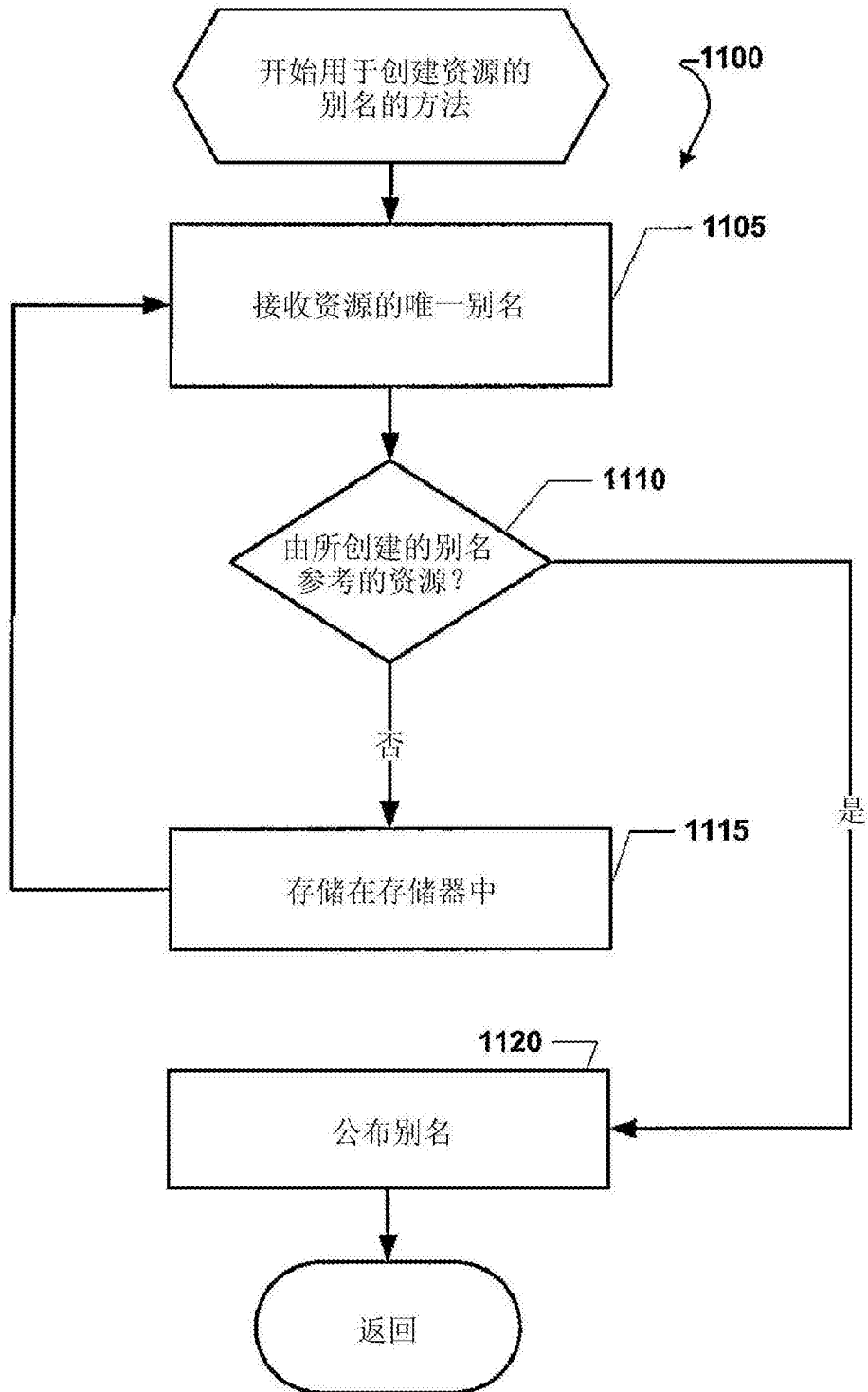


图11

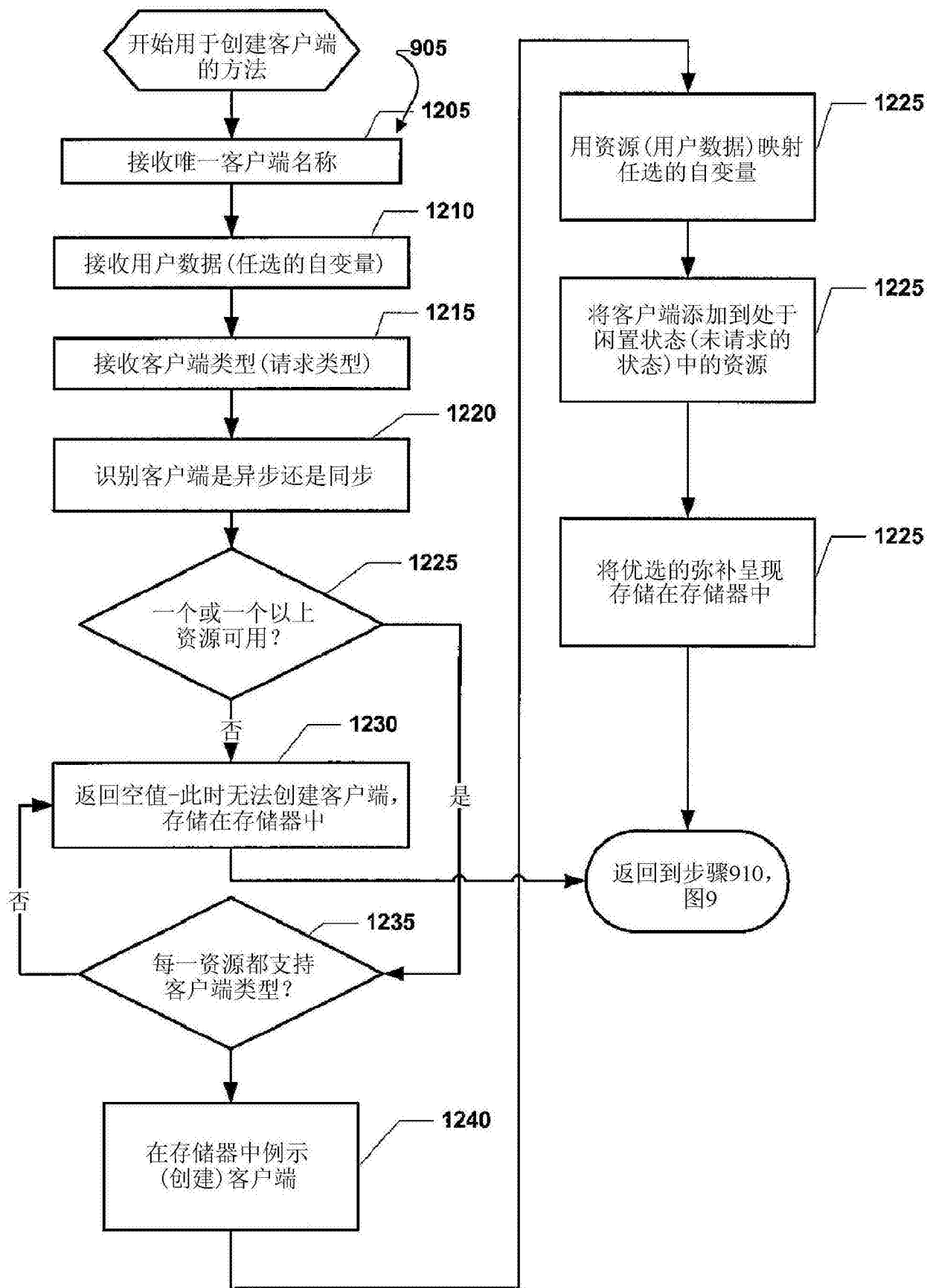


图12

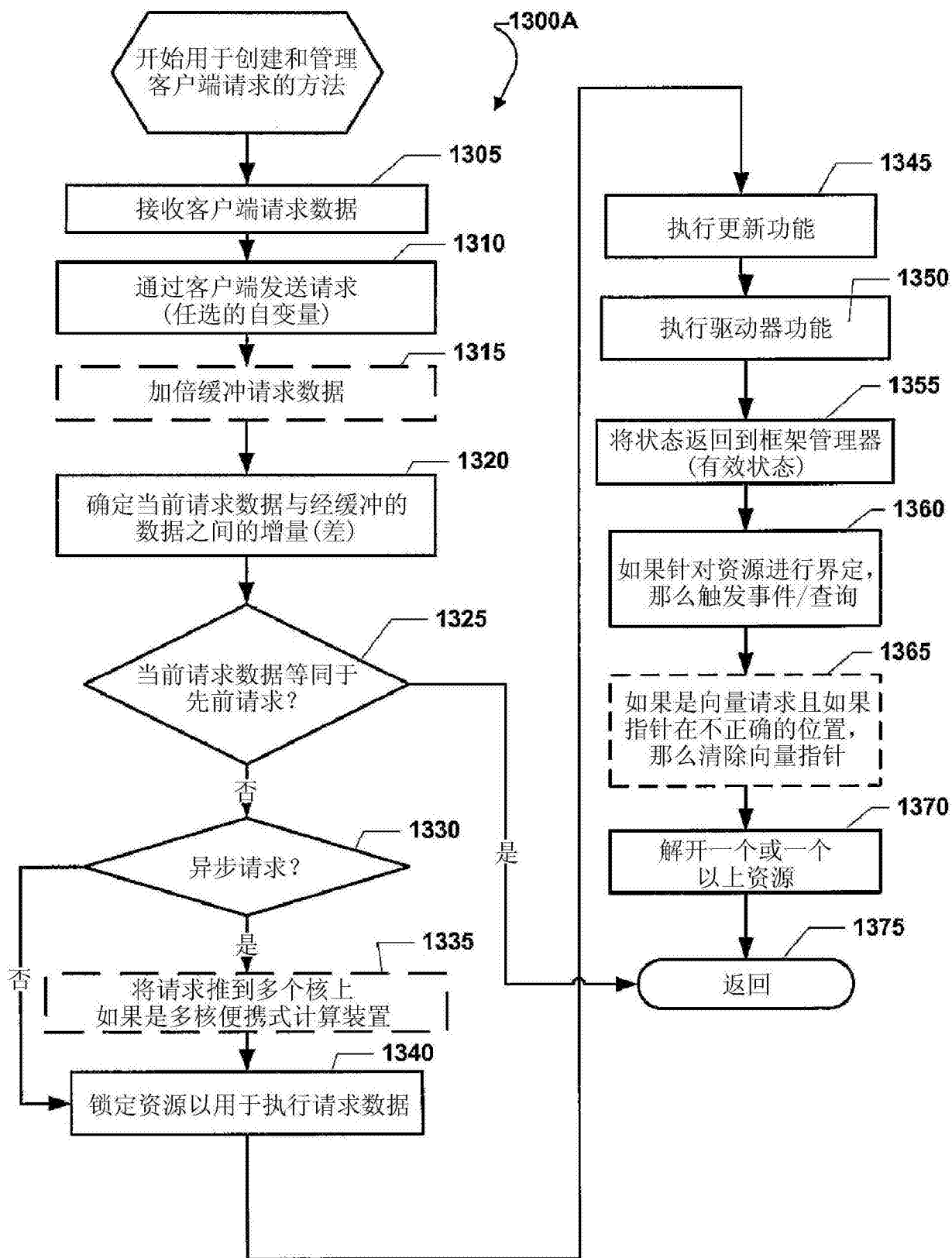


图13

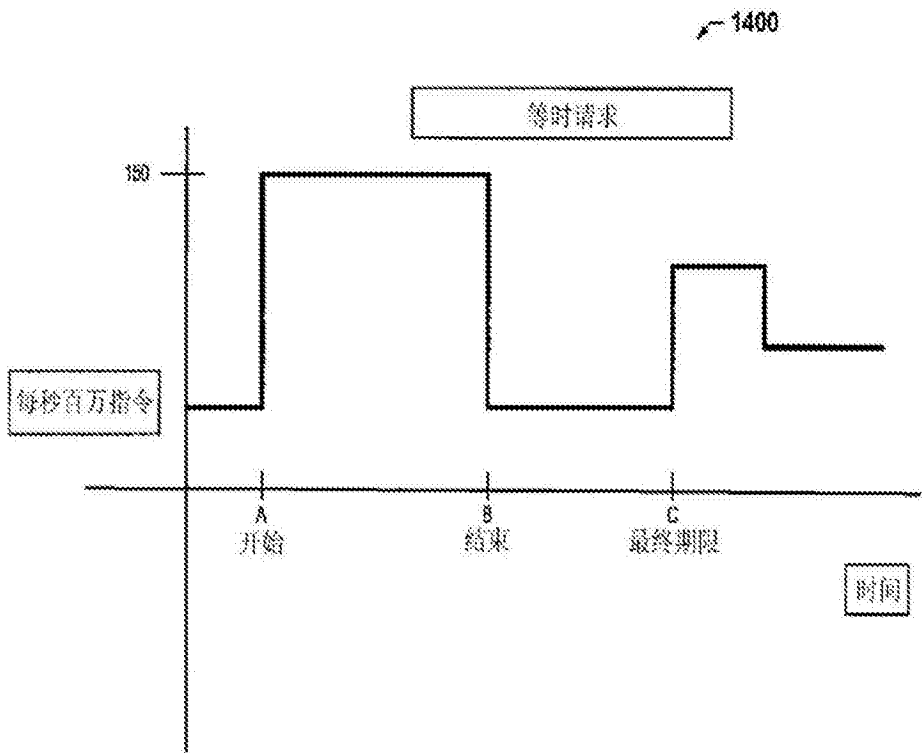


图14

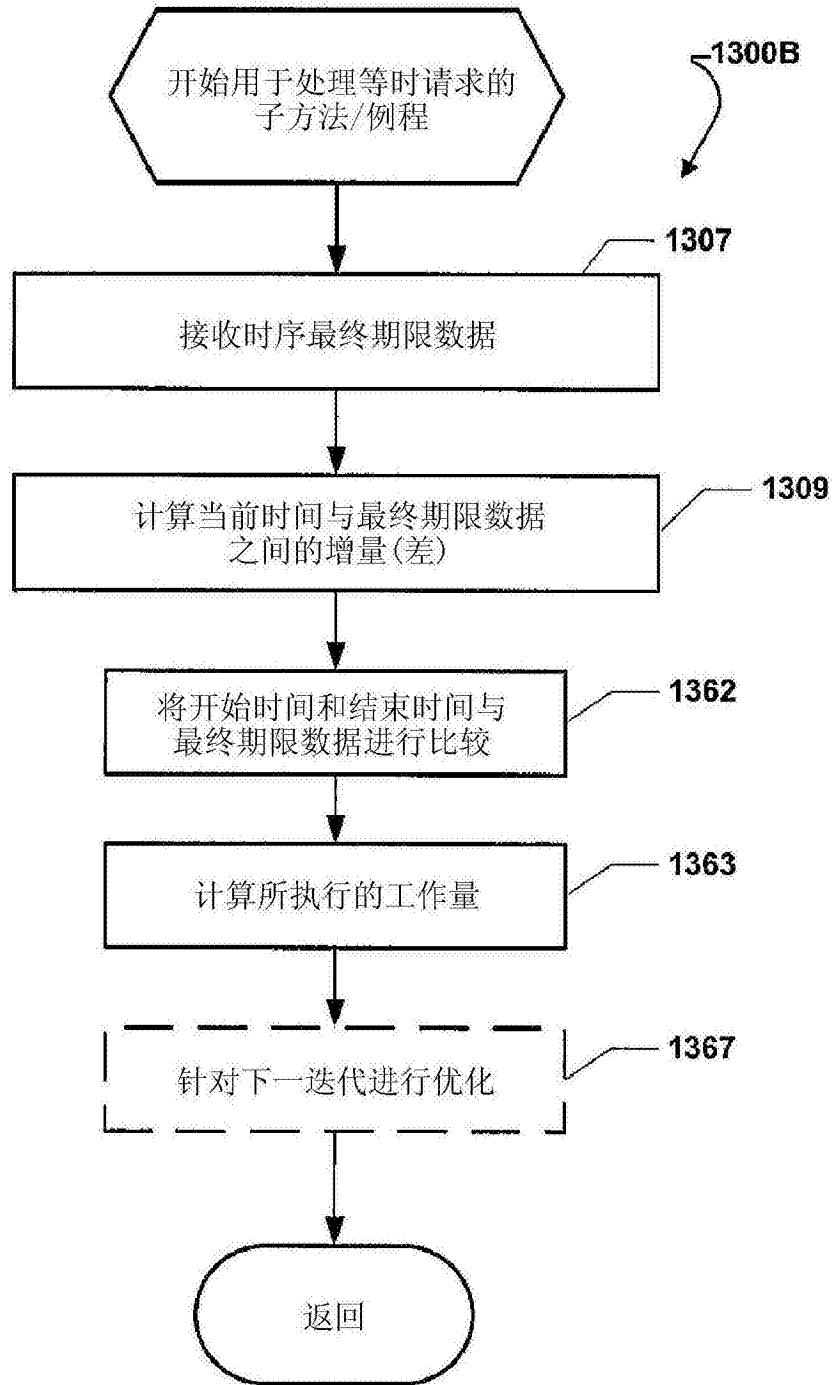


图15

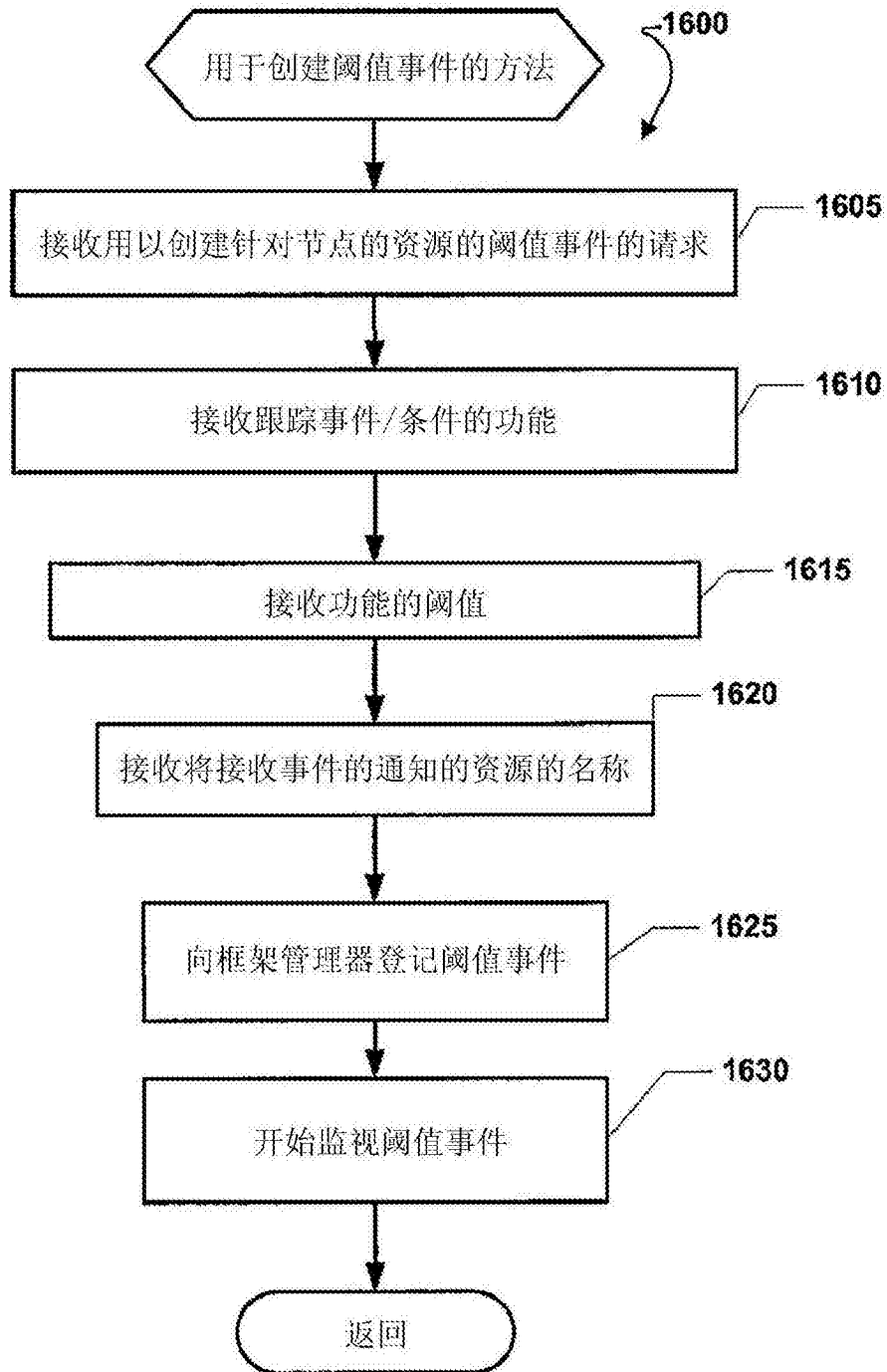


图16



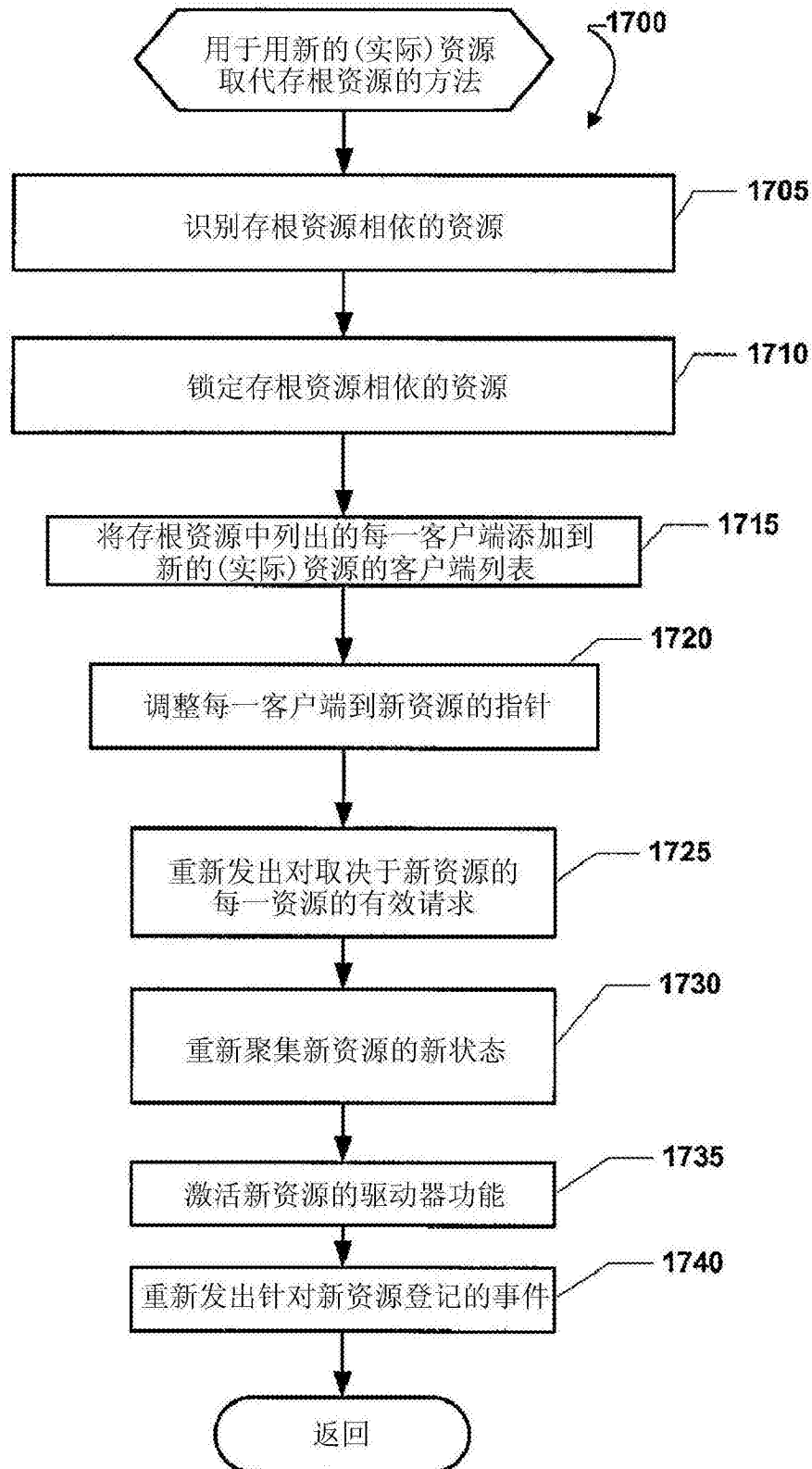


图17