(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 104969222 A
(43)申请公布日 2015.10.07

(21)申请号 201480005866.X
(22)申请日 2014.01.17
(30)优先权数据
13/748,173 2013.01.23 US
(85)PCT国际申请进入国家阶段日
2015.07.23
(86)PCT国际申请的申请数据
PCT/US2014/012121 2014.01.17
(87)PCT国际申请的公布数据
WO2014/116527 EN 2014.07.31
(71)申请人 脸谱公司
地址 美国加利福尼亚
(72)发明人 菲利普·刘 维萨·卡图里亚
(74)专利代理机构 北京康信知识产权代理有限责任公司 11240
代理人 梁丽超 陈鹏
(51)Int.Cl.
G06F 17/30(2006.01)
G06F 15/16(2006.01)

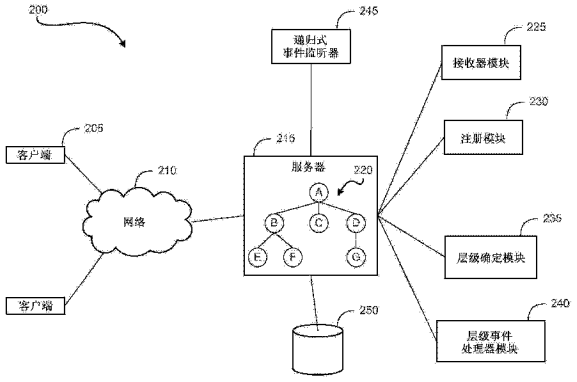
权利要求书3页 说明书11页 附图7页

(54)发明名称

使用层次数据结构节点上递归式事件监听器的方法和系统

(57)摘要

公开了一种用于在层次数据结构中的节点上注册递归监控器的方法和系统。本公开技术的实施方式可包括:(i)接收在源节点上注册事件监听器的请求,源节点是层级中彼此相关的多个节点中的一个;(ii)在源节点上注册事件监听器,事件监听器被配置为向客户端通知源节点中发生第一事件;以及(iii)如果源节点在层级中具有后代节点,将事件监听器设置为向客户端通知后代节点中发生第二事件,而不需要在后代节点上注册另一事件监听器。例如,每个节点均可代表存储设备的逻辑分区。



CN 104969222 A

1. 一种方法，包括：

在计算机网络中的计算机系统处，从所述计算机系统中的客户端接收在源节点上注册事件监听器的请求，所述源节点是层级中彼此相关的多个节点中的一个节点，每个所述节点均代表所述计算机系统中的存储设备的逻辑分区；

由所述计算机系统中的服务器在所述源节点上注册所述事件监听器，其中，所述事件监听器被配置为向所述客户端通知所述源节点中发生第一事件；

由所述服务器确定所述源节点在所述层级中是否具有后代节点；并且

响应于确定所述源节点在所述层级中具有后代节点，由所述服务器将所述事件监听器设置为向所述客户端通知在所述后代节点中发生第二事件，而不需要在所述后代节点上注册另一事件监听器。

2. 根据权利要求 1 所述的方法，进一步包括：

在所述源节点中所述第一事件或者在所述后代节点中所述第二事件中的至少一个发生时，向所述客户端通知所述第一事件或者所述第二事件中的所述至少一个发生。

3. 根据权利要求 2 所述的方法，其中，在所述源节点上注册所述事件监听器包括：

在向所述客户端通知发生所述第一事件或者所述第二事件之后，使所述事件监听器存留，并且

使所述事件监听器存留，直至由所述客户端移除所述事件监听器。

4. 根据权利要求 2 所述的方法，其中，向所述客户端通知发生所述第一事件或者所述第二事件包括：

监听所述第一事件的发生，其中，监听所述第一事件的发生包括由所述服务器确定是否将所述服务器的内存中的对应于所述源节点的字段设置为指示在所述源节点上注册所述事件监听器的预定值，

监听所述第二事件的发生，而不需要确定是否将所述服务器的内存中的对应于所述源节点的字段设置为指示在所述后代节点上注册所述事件监听器的预定值，并且

向所述客户端通知在所述源节点中发生所述第一事件或者在所述后代节点中发生所述第二事件。

5. 根据权利要求 1 所述的方法，其中，在所述源节点上注册所述事件监听器包括：将所述事件监听器配置为向所述客户端通知发生预定类型的所述第一事件或者所述第二事件。

6. 根据权利要求 5 所述的方法，其中，所述预定类型的所述第一事件或者所述第二事件包括以下各项中的至少一项：(i) 创建新节点，(ii) 删除现有节点，或者 (iii) 修改所述现有节点。

7. 根据权利要求 1 所述的方法，其中，所述第一事件和所述第二事件是相同预定类型的事件。

8. 根据权利要求 1 所述的方法，其中，所述第一事件和所述第二事件是不同的预定类型的事件。

9. 根据权利要求 1 所述的方法，其中，所述存储设备包括多个逻辑分区，各个所述逻辑分区均包含所述存储设备中包含的数据的单独非重叠子集。

10. 根据权利要求 1 所述的方法，其中，将所述事件监听器设置为向所述客户端通知在所述后代节点中发生所述第二事件包括：将所述事件监听器配置为向所述客户端通知在所

述层级中沿着自所述源节点开始的选定路径的后代节点中发生的事件。

11. 根据权利要求 1 所述的方法，其中，在所述源节点上注册所述事件监听器包括：通过配置访问控制将在所述源节点上所述事件监听器的注册限制为所述计算机系统中的用户角色的预定集合。

12. 一种方法，包括：

由计算机网络中的计算机系统确定在所述计算机系统的服务器中的源节点上是否发生了事件，所述源节点是层级中彼此相关的多个节点中的一个，每个所述节点均代表所述计算机系统中的存储设备的逻辑分区；并且

响应于确定在所述源节点中发生了所述事件，

由所述服务器确定是否由所述计算机系统中的客户端在所述源节点上注册事件监听器，

响应于确定在所述源节点上注册所述事件监听器，向所述客户端通知所述源节点中发生所述事件，及

响应于确定未在所述源节点上注册所述事件监听器；

由所述服务器确定所述源节点在所述层级中是否具有上代节点，

响应于确定所述源节点在所述层级中具有上代节点；

由所述服务器确定是否在所述上代节点上注册所述事件监听器，和

响应于确定在所述上代节点上注册所述事件监听器，向所述客户端通知在所述源节点中发生所述事件。

13. 根据权利要求 12 所述的方法，其中，即使在向所述客户端通知发生所述事件的动作之后，仍使在所述源节点上注册的所述事件监听器存留。

14. 根据权利要求 12 所述的方法，其中，所述事件包括以下各项中的至少一项：(i) 创建新节点，(ii) 删除现有节点，或者 (iii) 修改所述现有节点中。

15. 根据权利要求 12 所述的方法，进一步包括：

响应所述服务器与所述客户端之间的通信失败，从所述客户端接收在新服务器中的所述源节点上注册新事件监听器的请求，所述新服务器具有所述层级中的所述节点的至少一部分的副本，并且所述请求包括从所述服务器接收的最新事件的交易编号；

由所述新服务器并且使用所述交易编号获得自接收所述最新事件时起在所述源节点和所述源节点的后代中发生的事件的列表；并且

由所述新服务器将所述事件的列表发送给所述客户端。

16. 一种装置，包括：

服务器；

接收器模块，被配置为与所述服务器设备协作，从客户端接收在源节点上注册事件监听器的请求，所述源节点是层级中彼此相关的多个节点中的一个，每个所述节点均代表存储单元的逻辑分区；

注册模块，被配置为与接收器模块协作，在所述源节点上注册所述事件监听器，所述事件监听器被配置为向所述客户端通知所述源节点中发生第一事件；

层级确定模块，被配置为与所述注册模块协作，用于确定所述源节点在所述层级中是否具有后代节点；并且

层级事件处理器模块，被配置为与所述层级确定模块协作，响应于确定所述源节点在所述层级中具有后代节点，所述事件监听器向所述客户端通知在所述后代节点中发生第二事件，而不需要在所述后代节点上注册另一事件监听器。

17. 根据权利要求 16 所述的装置，其中，所述事件监听器被进一步配置为与所述注册模块协作，在所述源节点中所述第一事件或者在所述后代节点中所述第二事件中的至少一个发生时，向所述客户端通知所述第一事件或者所述第二事件中的所述至少一个发生。

18. 根据权利要求 17 所述的装置，其中，所述事件监听器模块被进一步配置为：

监听所述第一事件的发生，其中，监听所述第一事件的发生包括由所述服务器确定是否将所述服务器的内存中的对应于所述源节点的字段设置为指示在所述源节点上注册所述事件监听器的预定值，

监听所述第二事件的发生，而不需要确定是否将所述服务器的内存中的对应于所述源节点的字段设置为指示在所述后代节点上注册所述事件监听器的预定值，并且

向所述客户端通知在所述源节点中发生所述第一事件或者在所述后代节点中发生所述第二事件。

19. 根据权利要求 16 所述的装置，其中，所述事件监听器被进一步配置为通知发生预定类型的事件，所述预定类型的事件包括以下各项中的至少一项：(i) 创建新节点，(ii) 删除现有节点，或者 (iii) 修改所述现有节点。

20. 根据权利要求 16 所述的装置，其中，所述存储单元包括多个逻辑分区，各个所述逻辑分区均包含所述存储单元中包含的数据的单独非重叠子集。

使用层次数据结构节点上递归式事件监听器的方法和系统

[0001] 相关申请的交叉引用

[0002] 本申请要求于 2013 年 1 月 23 日提交的美国专利申请号 13/748,173 的优先权，通过引用将其全部内容结合在此。

技术领域

[0003] 本发明整体涉及计算机应用中的事件监听器。更具体地，本发明涉及使用层次数据结构中的节点上的递归式事件监听器。

背景技术

[0004] 在与计算机相关的技术中，事件监听器是以下元件：检测（“监听”）软件对象（以下称之为“对象”）中特定类型（或多个特定类型）的事件的发生，并且向订户通知事件发生。在具有处于层级的不同等级 (level) 的对象的层次数据结构中，为了检测任一对象事件的发生（诸如，创建对象、删除对象、或者修改对象等），目前的事件监听器技术在层次数据结构中的每个对象上注册事件监听器。这些技术在注册事件监听器时通常消耗大量的时间和内存。

[0005] 例如，考虑其中社交网络环境拥有 100000 个用户（客户端）以及包含按照代表某些用户信息的层次数据结构组织的 1 亿对象的服务器的情景。假定在对象上注册事件监听器消耗一比特内存。如果有 100000 个客户端，并且 100000 个客户端中的每个客户端均希望将事件监听器置于服务器的 1 亿个对象的每个上，则仅仅对于注册事件监听器就消耗 1.25 兆兆位内存。可以在对象上注册三种类型的事件监听器：(a) 对象创建；(b) 对象修改；以及 (c) 对象的子代。因此，仅仅对于注册事件监听器就消耗了 3.75 兆兆位的内存。此外，如果客户端针对层级中的每个对象发送单独的注册请求，则可能消耗大量的网络带宽和时间用于注册。至少就消耗的时间、内存、以及网络带宽而言，目前的事件监听器技术通常效率较低。而且，与注册事件监听器的当前技术相关联的这种低效率可随着客户端的数目、层级中的对象的数目、或这两者的增加而显著上升。在分布式系统架构中，新的分布式应用或者服务以及希望消费该服务的新客户端可能连续增加。在该分布式环境中，用于注册事件监听器的目前技术通常由于消耗大量的资源用于注册事件监听器而使整个系统的效率明显降低。

发明内容

[0006] 下面描述的是一种用于在层次数据结构中的节点上注册递归式事件监听器的技术。递归式事件监听器是一种事件监听器，其监听在其中注册递归式事件监听器的节点中发生的事件并且还监听在该节点的下代节点中发生的事件，而不需要在下代节点上注册递归式事件监听器或者任何其他事件监听器。尤其在具有层次数据结构的环境方面，该技术更为有效地利用了内存和处理资源，并且不受所监控的对象数目或者请求监控的客户端的数目的明显不利影响。

[0007] 本公开技术的实施方式包括从客户端接收在源节点上注册事件监听器的请求。源节点是层级中彼此相关的多个节点中的一个并且每个节点均代表了存储设备的逻辑分区。服务器在源节点上注册事件监听器并且事件监听器被配置为向客户端通知源节点中发生第一事件。此外，服务器确定源节点在层级中是否具有下代节点并且响应源节点在层级中具有下代节点的确定，服务器将事件监听器设置为向客户端通知下代节点中发生第二事件，而不需要在下代节点上注册另一事件监听器。

[0008] 在本公开技术的至少一些实施方式中，在该节点中发生第一事件或者下代节点中发生第二事件中的至少一个时，向客户端通知发生第一事件或者第二事件中的至少一个。

[0009] 在本公开技术的至少一些实施方式中，即使向客户端通知发生第一事件或者第二事件之后，在该节点上注册的事件监听器也存留。

[0010] 本发明的一些实施方式具有除上述所述之外或者替代上述所述的其他的方面、元件、特征、以及步骤。在本说明书的其余部分中描述了这些潜在的附加和替换。

[0011] 在涉及方法、存储介质、以及系统的附加权利要求中具体地公开了根据本发明的实施方式，其中，在另一权利要求类别（例如，系统）中也可以要求保护在一个权利要求类别（例如，方法）中提及的任何特征。

[0012] 在根据本发明的实施方式中，一种方法包括：

[0013] 在计算机网络中的计算机系统处，接收来自计算机系统中的客户端的、在源节点上注册事件监听器的请求，该源节点是层级中彼此相关的多个节点中的一个，每个节点均代表了计算机系统中的存储设备的逻辑分区；

[0014] 由计算机系统中的服务器在源节点上注册事件监听器，其中，事件监听器被配置为向客户端通知源节点中发生第一事件；

[0015] 由服务器确定源节点在层级中是否具有下代节点；并且

[0016] 响应于源节点在层级中具有下代节点的确定，由服务器将事件监听器设置为向客户端通知下代节点中发生第二事件，而不需要在下代节点上注册另一事件监听器。

[0017] 在进一步的实施方式中，该方法包括：

[0018] 在源节点中发生第一事件或者下代节点中发生第二事件中的至少一个时，向客户端通知发生第一事件或者第二事件中的至少一个。

[0019] 在源节点上注册事件监听器可包括：

[0020] 即使在向客户端通知发生第一事件或者第二事件之后，也使事件监听器存留；并且

[0021] 使事件监听器存留，直至客户端移除事件监听器。

[0022] 将向客户端通知发生第一事件或者第二事件，可包括：

[0023] 监听第一事件的发生，其中，监听第一事件的发生包括由服务器确定是否将对应于源节点的服务器的内存中的字段设置为指示在源节点上注册的事件监听器的预定值；

[0024] 监听第二事件的发生，而不需要确定是否将对应于源节点的服务器的内存中的字段设置为指示在下代节点上注册的事件监听器的预定值；并且

[0025] 将向客户端通知源节点中第一事件的发生或者下代节点中第二事件的发生。

[0026] 在源节点上注册事件监听器可包括将事件监听器配置为向客户端通知预定类型的第一事件或者第二事件的发生。

[0027] 预定类型的第一事件或者第二事件可包括以下各项中的至少一项(i)新节点创建、(ii)删除现有节点、或者(iii)修改现有节点。

[0028] 第一事件和第二事件可以为相同的预定类型的事件。

[0029] 第一事件和第二事件可以为不同的预定类型的事件。

[0030] 存储设备可包括多个逻辑分区,每个逻辑分区均包含存储设备中包含的数据的单独非重叠子集。

[0031] 将事件监听器设置为向客户端通知下代节点中第二事件的发生,可包括:将事件监听器配置为向客户端通知沿着从层级中的源节点起始的选择路径的下代节点中发生的事件。

[0032] 在源节点上注册事件监听器可包括:通过配置访问控制,将在源节点上注册的事件监听器限制为计算机系统中的用户角色的预定集合。

[0033] 在根据本发明的进一步实施方式中,一种方法包括:

[0034] 由计算机网络中的计算机系统确定计算机系统的服务器的源节点中是否发生了事件,源节点是层级中彼此相关的多个节点中的一个,每个节点均代表计算机系统中的存储设备的逻辑分区;并且

[0035] 响应于确定源节点中发生了事件;

[0036] 由服务器确定是否通过计算机系统中的客户端在源节点上注册事件监听器;

[0037] 响应于确定在源节点上注册事件监听器,向客户端通知源节点中事件的发生;并且

[0038] 响应于确定未在源节点上注册事件监听器;

[0039] 由服务器确定源节点在层级中是否具有上代节点;

[0040] 响应于确定源节点在层级中具有上代节点;

[0041] 由服务器确定是否在上代节点上注册事件监听器;并且

[0042] 响应于确定在上代节点上注册事件监听器,向客户端通知源节点中事件的发生。

[0043] 即使在向客户端通知事件发生的行为之后,在源节点上注册的事件监听器仍可存留。

[0044] 事件可包括以下各项中至少一个(i)创建新节点、(ii)删除现有节点、或者(iii)修改现有节点中。

[0045] 在进一步实施方式中,该方法包括:

[0046] 响应于服务器与客户端之间的通信失败,从客户端接收在新服务器中的源节点上注册新的事件监听器的请求,新服务器具有层级中的节点的至少一部分的副本,并且该请求包括从服务器接收的最新事件的交易编号;

[0047] 由新服务器并且使用交易编号获得自接收最新事件时起在源节点和源节点的下代中发生的事件的列表;并且

[0048] 由新服务器将事件的列表发送至客户端。

[0049] 在本发明的实施方式中,还要求保护一种装置,包括:

[0050] 服务器;

[0051] 接收器模块,被配置为与服务器设备协作,以从客户端接收在源节点上注册事件监听器的请求,源节点是层级中连接至彼此的多个节点中的一个,每个节点均代表存储单

元的逻辑分区；

[0052] 注册模块，被配置为与接收器模块协作，以在源节点上注册事件监听器，事件监听器被配置为向客户端通知源节点中第一事件的发生；

[0053] 层级确定模块，被配置为与注册模块协作，用于确定源节点在层级中是否具有下代节点；以及

[0054] 层级事件处理器模块，被配置为与层级确定模块协作，以响应于确定源节点在层级中具有下代节点，将事件监听器设置为向客户端通知下代节点中第二事件的发生，而不需要在下代节点上注册另一事件监听器。

[0055] 事件监听器可被进一步配置为与注册模块协作，以在源节点中发生第一事件或者下代节点中发生第二事件中的至少一个时，将向客户端通知发生第一事件或者第二事件中的至少一个。

[0056] 事件监听器模块可被进一步配置为：

[0057] 监听第一事件的发生，其中，监听第一事件的发生包括：由服务器确定是否将对应于源节点的服务器的内存中的比特设置为指示在源节点上注册事件监听器的预定值；

[0058] 监听第二事件的发生，而不需要确定是否将对应于源节点的服务器的内存中的比特设置为指示在后代节点上注册事件监听器的预定值；并且

[0059] 将向客户端通知源节点中发生的第一事件或者后代节点中发生的第二事件。

[0060] 事件监听器可被进一步配置为通知发生了预定类型的事件，预定类型的事件包括(i)创建新的节点、(ii)删除现有节点、或者(ii)修改现有节点中的至少一个。

[0061] 存储单元可包括多个逻辑分区，每个逻辑分区均包含存储单元中包含的数据的单独非重叠子集。

[0062] 在本发明的进一步实施方式中，一种或者多种计算机可读的非易失性存储介质体现在被执行时可操作为执行根据本发明或者上述所述实施方式中任一种所述的方法的软件。

[0063] 在本发明的进一步实施方式中，一种系统包括：一个或者多个处理器；和内存，内存耦接至包括由处理器执行的指令的处理器，处理器在执行指令时可操作为执行根据本发明或者上述所述实施方式中任一种所述的方法。

附图说明

[0064] 图1示出了本发明的实施方式可在其中运行的环境。

[0065] 图2是示出了层次数据结构中的节点上的递归式事件监听器的框图。

[0066] 图3是用于在层次数据结构中的节点上注册递归式事件监听器的流程图。

[0067] 图4是用于通知层次数据结构的节点中是否发生事件的流程图。

[0068] 图5提供示出了在图1的协调服务中注册递归式事件监听器的实施例。

[0069] 图6是示出了响应于客户端与服务器之间的通信失败而由客户端重新注册递归式事件监听器的过程的流程图。

[0070] 图7是可执行各种操作并且存储通过该操作产生和/或使用的各种信息的装置的框图。

具体实施方式

[0071] 在本说明书中，参考“实施方式”、“一种实施方式”等指本发明的至少一种实施方式中包括所描述的具体特征、功能、或者特性。在本说明书中出现的该等短语并不一定必须全部指同一实施方式，也不一定必须相互排斥。

[0072] 公开了一种用于在层次数据结构中的节点上注册递归式事件监听器的方法和系统。本公开技术的实施方式包括从客户端接收在源节点上注册事件监听器的请求。源节点是层级中彼此相关的多个节点中的一个并且每个节点均代表（例如）存储设备的逻辑分区。服务器在源节点上注册事件监听器，并且事件监听器被配置为向客户端通知源节点中发生第一事件。此外，服务器确定源节点在层级中是否具有后代节点并且响应于确定源节点在层级中具有后代节点，服务器将事件监听器设置为向客户端通知后代节点中发生第二事件，而不需要在后代节点上注册另一事件监听器。

[0073] 此后，除非另有注释，否则，术语“事件监听器”也指“递归式事件监听器”。非递归式事件监听器被称之为“非递归式事件监听器”。

[0074] 环境

[0075] 图 1 示出了本公开技术的实施方式可在其中运行的环境 100。如图所示，环境 100 包括协调服务 125，例如，协调服务可以是从马里兰，森林山 (Forest Hill, Maryland) 的阿帕奇软件基金会 (Apache Software Foundation) 获得的 ZooKeeper™服务的实现方式。在另一实施方式中，环境 100 可包括为分布式系统或者分布式应用提供协调服务的其他相似服务。协调服务 125 提供诸如用于分布式应用的同步、配置维护、分组、以及命名等协调服务。在称为集合体的一组服务器 105 上复制协调服务 125。（在下文中，服务器 120 的组 105 也被称之为“组”或者“集合体”）。只要组 105 中的大多数服务器可用，则可使用协调服务 125。组 105 中的每个服务器均具有诸如每个其他服务器的状态等信息。

[0076] 在集合体 105 中，一个服务器可用作主导器 115，主导器的任务是经由共识接受交易并且协调交易（诸如，写）。在该集合体中，除主导器 115 之外的服务器可用作跟随器，其是主导器 115 的直接、只读复制品。在协调服务 125 中，将来自客户端的一些写请求转发至主导器 115。跟随器从主导器 115 接收提议并且可同意或者不同意服务来自客户端的请求。此外，在协调服务 125 中，如果主导器 115 出现故障或者通过其他方式变得不可用，则任何其他跟随器可经由共识而变成新的主导器并且可继续服务请求。跟随器的其余部分可与新的主导器同步。

[0077] 环境 100 包括存储介质，诸如包含由使用协调服务 125 的应用（未示出）所使用的数据的数据库（未示出）。例如，在诸如社交网络环境的应用中，数据库可包含用户简档数据。数据库被划分成多个逻辑分区，也被称为“碎片”，其中每个逻辑分区均可包含数据库中的数据的子集。例如，在社交网络应用中，如果数据库包含 100000 个用户的用户简档数据，则第一碎片可包含用户 1 至 10000 的用户简档数据，第二碎片可包含用户 1001 至 20000 的用户简档数据等。数据库中的每个碎片均包含数据库中的数据的单独非重叠子集。

[0078] 数据库中的碎片可表示为层次数据结构（诸如树结构 130）中的节点的集合。例如，在服务器 120 的内存中创建树结构 130。客户端 110 通过访问服务器 120 的树结构 130 中的对应节点而访问数据库的碎片中的数据。树结构 130 中的根节点“A”可代表用户数据库，节点“B”可代表具有用户 1 至 10000 的用户数据的碎片，节点“C”可代表具有用户 10001

至 20000 的用户数据的碎片，并且节点“D”可代表具有用户 20001 至 30000 的用户数据的碎片。

[0079] 客户端 110 可订阅树结构 130 中的任一节点中发生事件的通知。为了使客户端 110 订阅节点中发生事件的通知，客户端 110 可请求服务器 120 在具体节点上注册事件监听器。例如，客户端 110 请求服务器 120 在树结构 130 中的节点 A 上注册事件监听器。事件监听器可以是递归式事件监听器或者非递归式事件监听器。

[0080] 非递归式事件监听器仅监控事件监听器注册在其上的节点中发生的事件。例如，如果客户端 110 已经在节点 A 上注册非递归式事件监听器，则仅在节点 A 中发生事件时才向客户端 110 通知发生事件。不向客户端 110 通知节点 A 的后代节点中发生的事件。（在一些实施方式中，如果节点的子代改变，则可通知客户端，但是，仅通知一次，例如，第一次。）另一方面，如果客户端 110 在节点 A 上注册递归式事件监听器，则向客户端 110 通知节点 A 或者在树结构 130 中的节点 A 的后代节点中发生的事件。例如，如果客户端 110 已经在节点 A 上注册递归式事件监听器，则向客户端 110 通知在节点 A 和 / 或节点 A 的任一后代节点（即，节点 B 至 F）中发生的事件。同样，如果在节点 B 上注册递归式事件监听器，则向客户端 110 通知在节点 B 或者其后代节点 E 和 F 中的任一个中发生的事件。

[0081] 因为在节点上注册递归式事件监听器将在该节点或者其后代节点中发生的事件通知给订户，所以本技术消除了使客户端在该节点的后代节点上注册事件监听器的需求。从而减少了在所有后代节点上注册非递归式事件监听器所消耗的时间和网络带宽。此外，因为可以存储用于监控节点及其后代节点的单个注册，所以还可减少用于存储递归式事件监听器注册所消耗的内存。另一方面，在非递归式注册中，通常，存在与客户端 100 希望监控的节点的数目一样多的注册。因此，至少就具有层次数据结构的应用中的目前非递归式事件监听器所消耗的时间、内存、以及网络带宽方面而言，递归式事件监听器允许明显改善整个系统的性能。

[0082] 图 2 是示出了根据本公开技术的实施方式的、层次数据结构中的节点上的递归式事件监听器的框图。系统 200 可以是图 1 中的环境 100 或者相似环境。系统 200 包括服务器 215 以及通过通信网络 210 与服务器 215 通信的客户端 205。客户端 205 和服务器 215 可在相同的物理机器或者不同的机器上运行。客户端 205 可请求在服务器 215 中的树结构 220 上注册递归式事件监听器 245。例如，客户端 205 请求在树结构 220 的节点 B 上注册递归式事件监听器 245。被配置为与服务器 215 通信的接收器模块 225 从客户端 205 接收在树结构 220 中的节点 B 上注册递归式事件监听器 245 的请求。

[0083] 注册模块 230 在树结构 220 中的节点 B 上注册递归式事件监听器 245。此外，注册模块 230 将服务器的内存中对应于节点 B 的字段（例如，比特或者比特组）设置为指示在节点 B 上注册递归式事件监听器 245 的预定值。在节点 B 上注册的递归式事件监听器 245 向客户端 205 通知在节点 B 中发生事件。在节点 B 上注册递归式事件监听器 245 之后，层次确定模块 235 确定节点 B 在树结构 220 中是否具有任何后代节点。响应于确定节点 B 具有后代节点 E 和 F，层级事件处理器模块 240 将在节点 B 上注册的递归式事件监听器 245 设置成监听并且通知在除节点 B 之外的节点 E 或者 F 中所发生的事件。递归式事件监听器 245 将在树结构 220 中的节点 B、E、或者 F 中的至少一个中发生的事件向客户端 205 通知。因此，递归式事件监听器 245 提供以下优点：获得在多个节点中发生的事件的通知、且仅在

单个节点上注册。

[0084] 能够在树结构 220 中的节点中发生的事件类型包括：创建新的节点、修改现有节点、或者删除现有节点。在节点 B、E 或者 F 中发生的事件可以是相同类型或者不同的类型。此外，事件监听器 245 被配置为将所发生的任意上述事件类型向客户端通知。

[0085] 在实施方式中，树结构 220 中的每个节点均代表存储单元（诸如数据库 250）的不同逻辑分区。在另一实施方式中，至少树结构 220 中的一些节点代表了数据库 245 的不同逻辑分区中包含的数据。例如，树结构 220 可代表在数据库的多个碎片中具有多个用户的用户简档数据的用户数据库。在又一实施方式中，节点可代表系统 200 的其他实体。树结构 220 可被包含在服务器 215 的内存中。

[0086] 各个模块，即，接收器模块 225、注册模块 230、层级确定模块 235、层级事件处理器模块 240、以及递归式事件监听器 245 可与服务器 215 在相同的机器上运行、或者在不同的机器上运行、可与服务器 215 集成、或者可与服务器 215 协作运行。

[0087] 客户端 205 还可以在节点 A 至 G 中的任意节点上注册非递归式事件监听器。在实施方式中，如果由递归式事件监听器和非递归式事件监听器两者监听节点中的事件，则仅将事件发生的一个通知发送至客户端 205。例如，如果客户端 205 已经在节点 B 上注册递归式事件监听器 245 并且在节点 B 或者其后代节点 E 和 F 上注册非递归式事件监听器（未示出），则在节点 B、E、或者 F 中发生事件时，客户端 205 将仅接收有关节点 B、E、或者 F 的任意节点中发生的事件的通知。客户端可能未从服务器接收有关递归式事件监听器和非递归式事件监听器的单独通知。

[0088] 即使在将节点 B、E、或者 F 中发生的事件向客户端 205 通知之后，在节点（例如，节点 B）上注册的递归式事件监听器 245 也存留。另一方面，在将该节点中发生的事件向客户端通知 205 之后，移除在节点上注册的非递归式事件监听器。不得不再次注册非递归式事件监听器，以继续从该节点获得发生的事件的通知。不同于非递归式事件监听器，不需要再次在该节点上注册递归式事件监听器 245，以继续获得该节点中发生的事件的通知。因此，与非递归式事件监听器相比较，至少就在节点上注册递归式事件监听器所消耗的时间和网络带宽方面而言，递归式事件监听器 245 是高效的。

[0089] 用于在层次数据结构中的节点上注册递归式事件监听器的方法

[0090] 图 3 是根据本公开技术的实施方式的、用于在层次数据结构中的节点上注册递归式事件监听器的过程 300 的流程图。在诸如图 2 的系统 200 的系统中可以实现过程 300。在步骤 305，服务器（或者与服务器协作工作的接收器模块）从客户端接收对于在源节点上注册事件监听器的请求。源节是层级中彼此相关的多个节点中的一个。层级中的每个节点均代表了数据库的不同逻辑分区。在步骤 310，服务器（或者与服务器协作工作的注册模块）在源节点上注册事件监听器并且将事件监听器配置为向客户端通知源节点中第一事件的发生。

[0091] 在确定步骤 315，服务器（或者与服务器协作工作的层级确定模块）确定源节点在层级中是否具有后代节点。响应于确定源节点在层级中不具有后代节点，过程 300 返回至 330。另一方面，响应于确定源节点在层级中具有后代节点，在步骤 320，服务器（或者与服务器协作工作的层级事件处理器模块）将事件监听器设置为向客户端通知后代节点中第二事件的发生，而不需要在后代节点上注册另一事件监听器。

[0092] 如步骤 325 所示，在源节点中发生第一事件或者后代节点中发生第二事件时，事件监听器向客户端通知至少一个第一事件或者第二事件的发生。

[0093] 图 4 是根据本公开技术的实施方式的、用于通知层次数据结构的节点中发生的事件的过程 400 的流程图。在诸如图 2 的系统 200 的系统可以实现过程 400。在步骤 405，服务器确定该服务器的节点中是否发生了事件。该节点是层级中彼此相关的多个节点中的一个。响应于确定该节点中未发生事件，过程在 435 返回。

[0094] 另一方面，响应于确定在该节点中发生了事件，在确定步骤 410，服务器确定是否通过客户端在节点上注册了事件监听器。响应于确定在该节点上注册了事件监听器的，在步骤 430，事件监听器向客户端通知该节点中发生事件。另一方面，响应于确定未在该节点上注册事件监听器，在确定步骤 415，服务器确定该节点在层级中是否具有上代节点。响应于确定该节点在层级中不具有上代节点，过程 400 在 435 返回。

[0095] 另一方面，响应于确定该节点在层级中确实具有上代节点，在确定步骤 420，服务器确定是否已在该上代节点上注册了事件监听器。响应于确定在上代节点上注册了事件监听器的，在步骤 430，事件监听器向客户端通知该节点中发生事件。另一方面，响应于确定未在上代节点上注册事件监听器，在确定步骤 425，服务器确定上代节点是否是层级的根节点。响应于确定上代节点不是层级的根节点，将对过程的控制转移至步骤 415。另一方面，响应于确定上代节点是层级的根节点，服务器意识到未在该节点或者上代节点上注册任何递归式事件监听器，进而过程 400 在 435 返回。

[0096] 从图中可以看出，仅在一个节点上注册的递归式事件监听器可提供通知在多个节点（即，递归式事件监听器在其上注册的节点及该节点的后代节点）中发生的事件的益处。

[0097] 操作实施例

[0098] 图 5 提供示出了根据本公开技术的实施方式的、使用诸如图 1 中所示的协调服务中的递归式事件监听器的实施例。协调服务 500 包括代表从用户数据库 510（作为层次数据结构 515）获得的用户简档数据的服务器 505。数据库 510 可包含多个用户的数据，例如，社交网络应用中的用户的数据。数据库 510 包括多个逻辑分区，即，LP1、LP2、…、至 LPn。每个逻辑分区均包含来自用户数据库 510 的用户数量可配置的的数据。在层次数据结构 515 中，用户数据库 510 可被视为根节点并且用户数据库 510 的逻辑分区可被视为根节点的后代节点。

[0099] 客户端可请求在层次数据结构 515 中的任意节点上注册递归式事件监听器、非递归式事件监听器、或者这两者。例如，考虑其中客户端 525 具有在根节点上注册的递归式事件监听器，用户数据库 510，客户端 530 具有在后代节点 LP2 上注册的递归式事件监听器和非递归式事件监听器，以及客户端 535 具有在根节点上注册的非递归式事件监听器，用户数据库 510 的情景。所有的事件监听器（即，递归式事件监听器和非递归式事件监听器）皆能够将发生的各种事件类型（诸如，在层次数据结构 515 中发生的创建、删除、或者修改等）中任意事件类型向客户端通知。

[0100] 例如，假定在层次数据结构 515 中发生创建新的逻辑分区（诸如 LP_{n+1}520）。因为客户端 525 的递归式事件监听器监听在根节点、用户数据库 510、以及后代节点中发生的事件，所以向客户端 525 通知新的后代节点 LP_{n+1}520 的创建。对于客户端 530，无论通过客户端 530 在后代节点 LP2 上注册的事件监听器是递归式或者非递归式，由于后代节点 520 的

创建既不是在节点LP2也不是在节点LP2的后代节点中发生的事件，所以不将后代节点520的创建向客户端530通知。对于客户端535，在根节点510上注册的非递归式事件监听器仅监听在节点510中发生的事件并且由此并不获知创建了新的后代节点520。因此，不将发生在层次数据结构中新的后代节点520的创建向客户端535通知。

[0101] 服务器505可提供用于注册递归式事件监听器的访问控制特征。例如，可将在根节点上注册事件监听器限制为一组用户或者用户角色，例如，管理员角色或者超级用户角色等。访问控制特征还可指定允许进行以下动作的一组用户或者用户角色(i)在任意节点上注册递归式事件监听器、(ii)注册递归式事件监听器用于特定事件类型、(iii)沿着层级中的特定路径注册递归式事件监听器等。

[0102] 此外，服务器还可提供用于递归式事件监听器的过滤器。过滤器可指定层次数据结构中将事件监听器设置为沿着其进行监听的路径，所监听的事件类型等。例如，在图2的树结构220中，可将过滤器应用于在根节点A上注册的递归式事件监听器，以仅监听在沿着路径A-B-F的节点中发生的事件。作为另一实例，过滤器可指定仅通知删除类型事件的发生。

[0103] 在一些实施方式中，诸如网络分区、客户端或者服务器故障等故障可致使客户端与服务器断开。服务器自动解注册与该客户端相关联的递归式事件监听器，并且客户端将停止从服务器接收事件通知。客户端可尝试在新的服务器上重新注册递归式事件监听器并且继续接收事件通知，其中包括在断开期间发生的事件。同样，客户端可假定其接收了在给定节点的层级下发生的所有事件，并且通过客户端上的应用所维持的节点结构与服务器端匹配。

[0104] 在没有这种机制的情况下，客户端可能不得不在重新连接之后并且在注册递归式事件监听器之前对整个层级进行检索，因为节点结构在故障期间可能已经发生改变。在至少一些实施方式中，给出了在基础数据库中对节点层级做出的每次更改（例如，创建、修改、删除）的交易编号。无论客户端何时从服务器中读取节点，其也会接收更改该节点的最新交易编号。当客户端尝试重新连接时，其可通过提供交易编号而在该节点上注册递归式事件监听器。服务器尝试传递自对应于所提供的交易编号的发生事件的所有发生事件的通知。

[0105] 服务器对节点层级进行遍历，以确定需要被传递至客户端的事件。此外，因为从遍历给定节点的当前层级不能确定删除节点，所以服务器参考（例如，作为日志存储在数据库中的）事件历史，以确定已删除的节点的列表。在至少一些实施方式中，根据相关联的交易编号将发送至客户端的事件列表分类。

[0106] 图6是示出了根据本公开技术的实施方式的、响应客户端与服务器之间的通信失败而通过客户端重新注册递归式事件监听器的过程600的流程图。在诸如系统200的系统中可以执行过程600。在确定步骤605，服务器确定给定的交易编号是否比预定的阈值时间旧。在至少一些实施方式中，服务器存储一预定时间段内的事件历史。如果服务器确定该交易比预定的阈值时间旧，则在步骤610，服务器拒绝注册递归式事件监听器的请求。在至少一些实施方式中，客户端可通过从请求中省去交易编号而重新尝试注册。

[0107] 另一方面，如果服务器确定该交易不比预定的阈值时间就，则在确定步骤615，服务器确定给定节点的层级是否具有任何未访问的节点，即，删除节点。如果存在任何未访问

的节点，则在步骤 645，服务器将有关各个未访问节点的删除事件添加到事件列表中。在至少一些实施方式中，服务器可通过读取数据库中存储的事件的日志（交易历史）而确定删除节点。另一方面，如果用户确定不存在删除节点，则在确定步骤 620，服务器确定在发生该交易（对应于给定的交易编号）之后是否创建了当前节点。

[0108] 如果服务器确定在该交易之后添加了当前节点，则在步骤 625，服务器将创建节点事件添加到列表中并且将控制转移至步骤 640。另一方面，如果服务器确定在该交易之后未添加当前节点，则在确定步骤 630，服务器确定是否在该交易之后修改了当前节点。如果服务器确定在该交易之后修改了当前节点，则在步骤 635，服务器将修改节点事件添加到列表中，并且将控制转移至步骤 640。在步骤 640，服务器对层级中的下一后代节点进行遍历。在至少一些实施方式中，服务器按照广度优先搜索顺序对层级进行遍历。在服务器对给定节点的整个层级进行遍历之后，在步骤 650，服务器根据事件的相关交易次数将事件列表分类，并且在步骤 655，服务器将事件列表传递至客户端。

[0109] 用于实现方式的装置

[0110] 图 7 是根据本公开技术的、可执行各种操作并且存储通过该操作产生和 / 或使用的各种信息的装置的框图。该装置可代表此处所描述的任意计算机。计算机 700 旨在示出硬件设备，在硬件设备上可以实现在图 1 至图 6 的实施例中所描述的任何实体、部件、或者服务（以及在本说明书中描述的任何其他部件），诸如，协调服务、服务器、客户端、数据库、树结构等。计算机 700 包括耦接至互连 703 的一个或者多个处理器 701 和内存 702。互连 703 在图 7 中被示为代表任意一个或者多个单独的物理总线、点到点连接、或者通过适当的网桥、适配器、或者控制器连接的物理总线和点到点连接的抽象表示法。因此，例如，互连 703 可包括系统总线、外围设备互连 (PCI) 总线或者 PCI-Express 总线、超传输或工业标准架构 (ISA) 总线、小型计算机系统接口 (SCSI) 总线、通用串行总线 (USB)、IIC(I2C) 总线、或者电学与电子工程师协会 (IEEE) 标准 1394 总线（也被称为“火线”）。

[0111] 处理器 701 是计算机 700 的中央处理单元 (CPU) 并因此控制计算机 700 的整个操作。在特定实施方式中，处理器 701 通过执行存储在内存 702 中的软件或者固件而实现此操作。处理器 701 可以是或者可包括一个或者多个可编程的通用或者专用微处理器、数字信号处理器 (DSP)、可编程控制器、专用集成电路 (ASIC)、可编程逻辑设备 (PLD)、可信平台模块 (TPM) 等、或者这种装置的组合。

[0112] 内存 702 是或者包括计算机 700 的主内存。内存 702 代表了任何形式的随机访问内存 (RAM)、只读内存 (ROM)、闪存等、或者这种装置的组合。在使用中，内存 702 可包含代码。在一种实施方式中，代码包括通用编程模块，通用编程模块被配置为组织经由计算机总线接口接收的通用程序并且制备用于在处理器执行的通用程序。在另一实施方式中，使用诸如 ASIC、PLD、或场可编程门阵列 (FPGA) 等硬连接电路可实现通用编程模块。

[0113] 网络适配器 707、存储设备 705、以及 I/O 设备 706 也通过互联 703 连接至处理器 701。网络适配器 707 向计算机 700 提供通过网络与远程设备通信的能力并且可以例如是以太网适配器或者光纤信道适配器。网络适配器 707 还可向计算机 700 提供与集群内的其他计算机通信的能力。在一些实施方式中，计算机 700 可使用多于一个的网络适配器分开地处理集群内和外的通信。

[0114] 例如，I/O 设备 706 可包括键盘、鼠标或者其他定点设备、盘驱动、打印机、扫描仪、

以及包括显示设备的其他输入和/或输出设备。例如，显示设备可包括阴极射线管(CRT)、液晶显示器(LCD)、或者一些其他适用的已知或者常规显示设备。

[0115] 内存702中存储的代码可被实现为将处理器701编程以完成上述动作的软件和/或固件。在特定实施方式中，最初，通过计算机700(例如，经由网络适配器707)从远程系统下载该软件或者固件可将该软件或者固件提供至计算机700。

[0116] 例如，通过利用软件和/或固件编程的、或者通过整个专用硬连接(非编程)电路、或者这种形式的组合中的可编程电路(例如，一个或者多个微处理器)可实现此处介绍的技术。例如，专用硬连接电路可以是一个或者多个ASIC、PLD、FPGA等的形式。

[0117] 在实现此处介绍的技术时所使用的软件或者固件可被存储在机器可读的存储介质中并且可由一个或者多个通用或专用可编程微处理器执行。此处所使用的术语“机器可读存储介质”包括能够以机器可访问的形式存储信息的任何机制。

[0118] 机器还可以是服务器计算机、客户端计算机、个人电脑(PC)、平板PC、膝上型电脑、机顶盒(STB)、个人数字助理(PDA)、蜂窝电话、iPhone、黑莓、处理器、电话、网络设备、网络路由器、交换机或者网桥、或者能够执行指定由该机器采取的动作的一组指令(序列或者其他)的任何机器。

[0119] 例如，机器可访问的存储介质或者存储设备705包括可记录/非可记录媒体(例如，ROM、RAM、磁盘存储媒体、光学存储媒体、闪存设备等)等、或者其任何组合。存储介质通常可以是非易失性的或者包括非易失性设备。在上下文中，非易失性存储介质可包括有形设备，即，设备具有具体的物理形式，尽管设备可改变其物理状态。因此，例如，非易失性指保持有形的设备，尽管物理状态改变。

[0120] 此处所使用的术语“逻辑的”例如可包括利用专用软件和/或固件编程的可编程电路、专用硬连接电路、或者其组合。

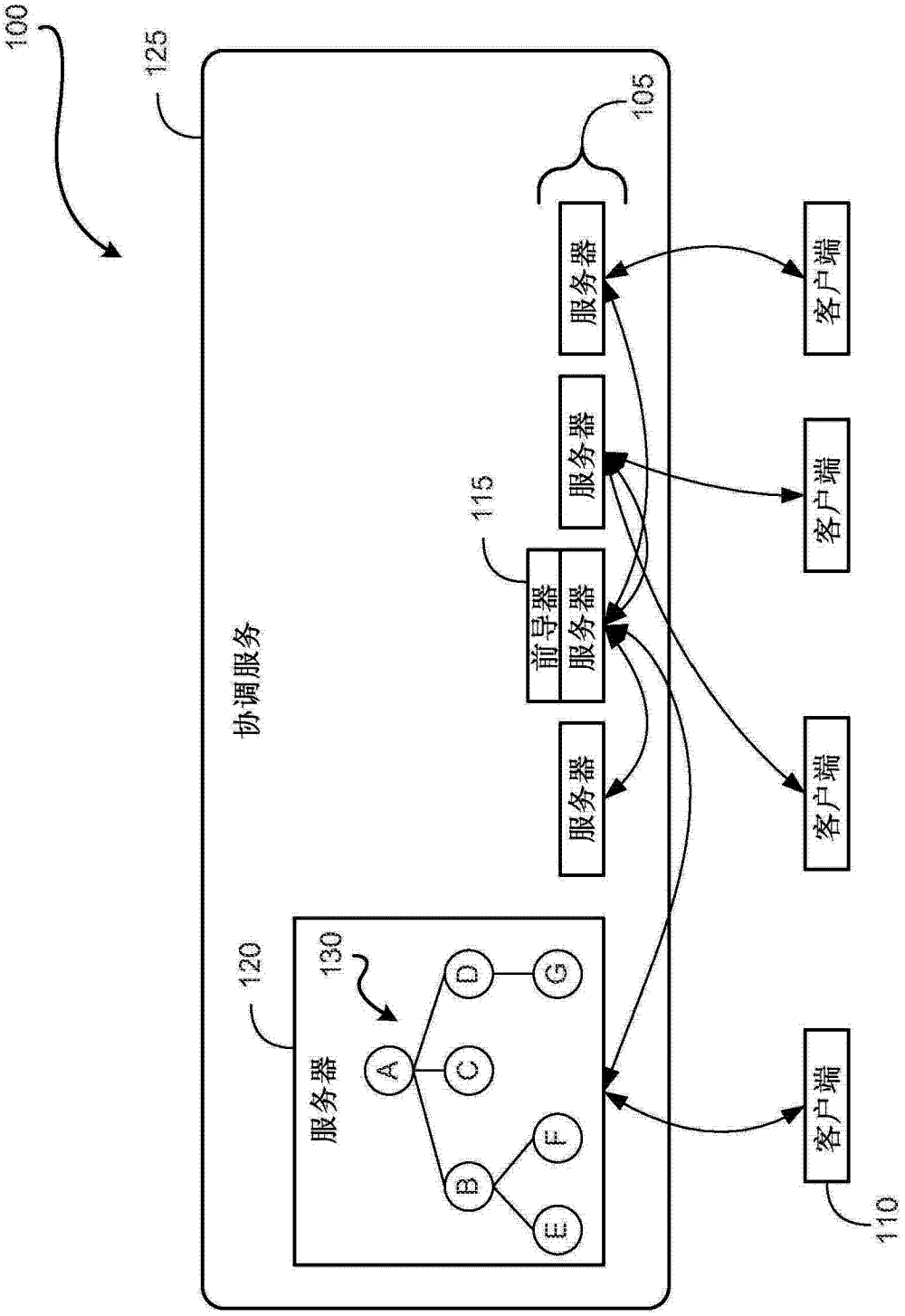

100
125
105
协调服务
服务器
服务器
115
前导器
服务器
服务器
客户端
客户端
客户端
客户端
120
130
服务器
A
B
C
D
E
F
G
110

图 1

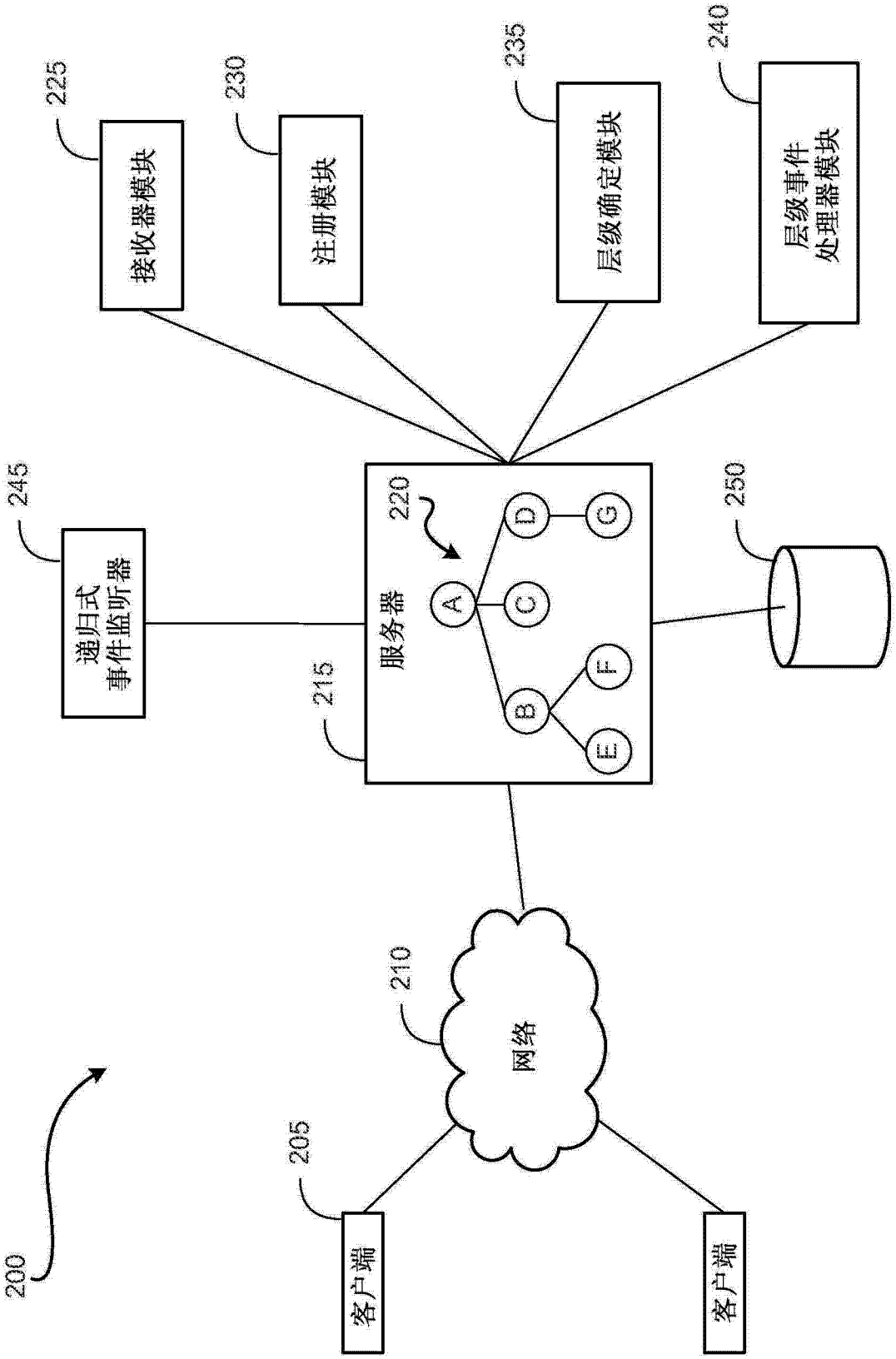

200
205
客户端
客户端
210
网络
215
服务器
220
A
B
C
D
E
F
G
225
接收器模块
230
注册模块
235
层级确定模块
240
层级事件
处理器模块
245
递归式
事件监听器
250

图 2

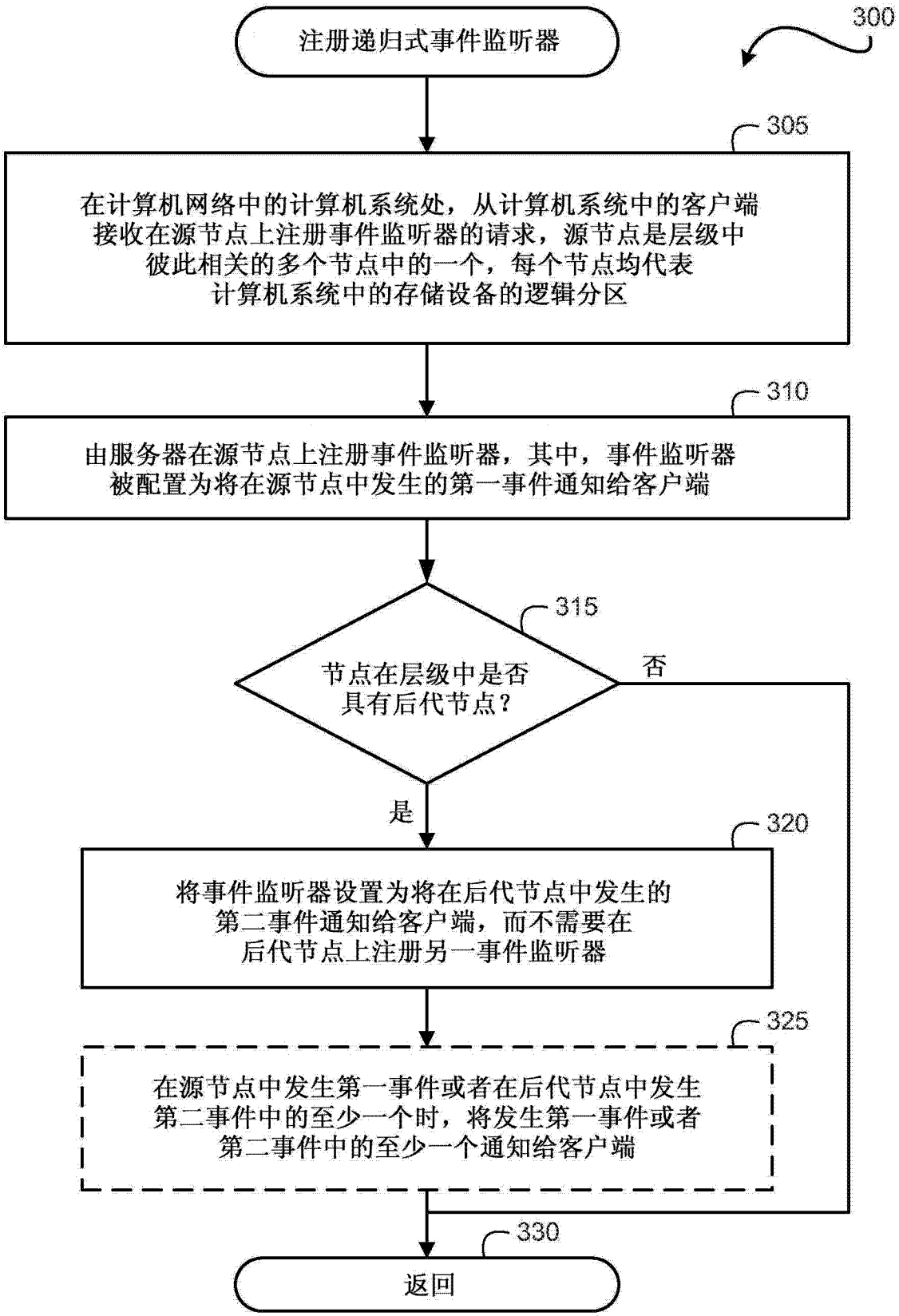

注册递归式事件监听器
300
305
在计算机网络中的计算机系统处，从计算机系统中的客户端接收在源节点上注册事件监听器的请求，源节点是层级中彼此相关的多个节点中的一个，每个节点均代表计算机系统中的存储设备的逻辑分区
310
由服务器在源节点上注册事件监听器，其中，事件监听器被配置为将在源节点中发生的第一事件通知给客户端
315
节点在层级中是否具有后代节点？
否
是
320
将事件监听器设置为将在后代节点中发生的第二事件通知给客户端，而不需要在后代节点上注册另一事件监听器
325
在源节点中发生第一事件或者在后代节点中发生第二事件中的至少一个时，将发生第一事件或者第二事件中的至少一个通知给客户端
330
返回

图 3

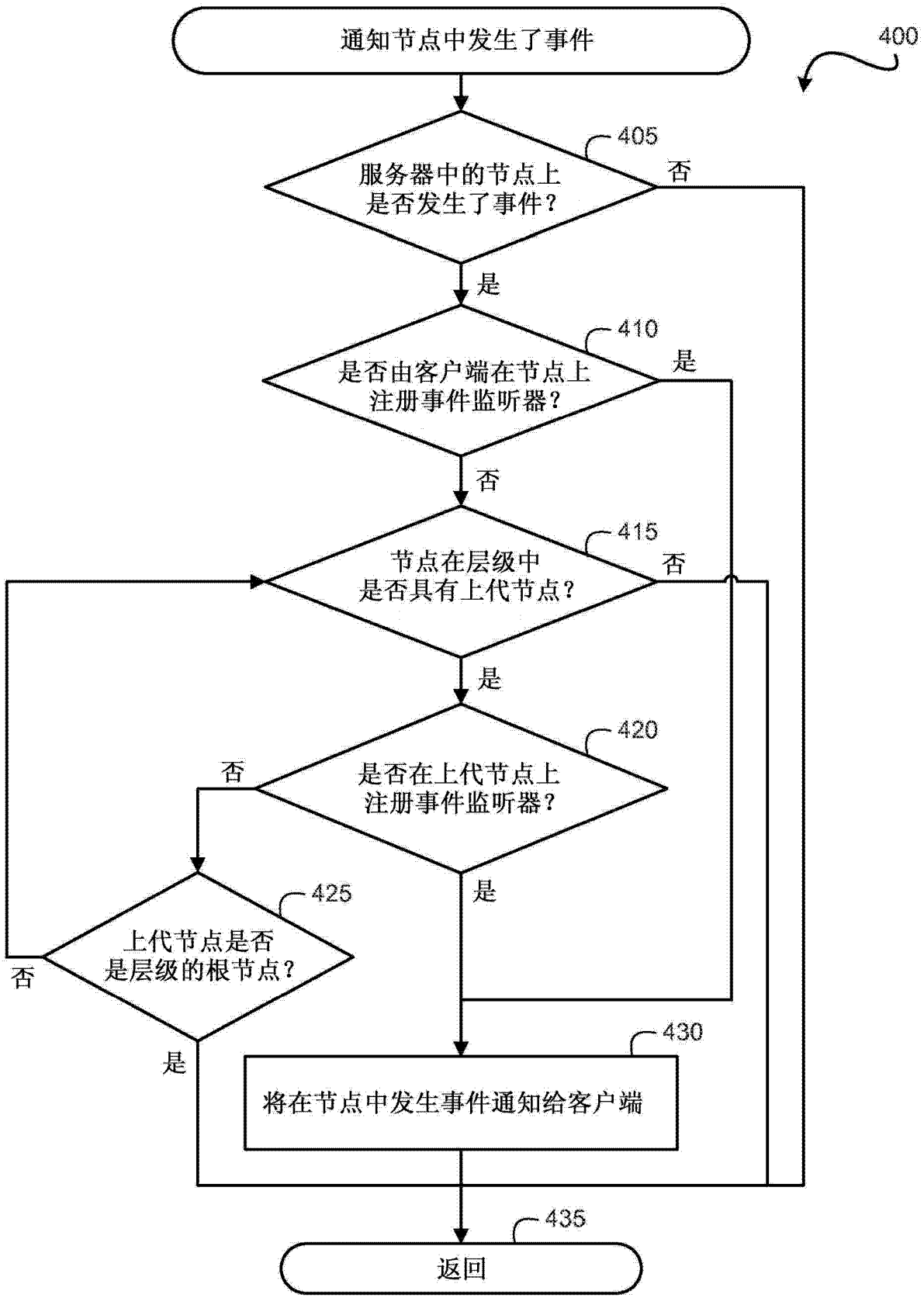

通知节点中发生了事件
400
405
服务器中的节点上
是否发生了事件？
否
是
410
是否由客户端在节点上
注册事件监听器？
是
否
415
节点在层级中
是否具有上代节点？
否
是
420
是否在上代节点上
注册事件监听器？
否
是
425
上代节点是否
是层级的根节点？
否
是
430
将在节点中发生事件通知给客户端
435
返回

图 4

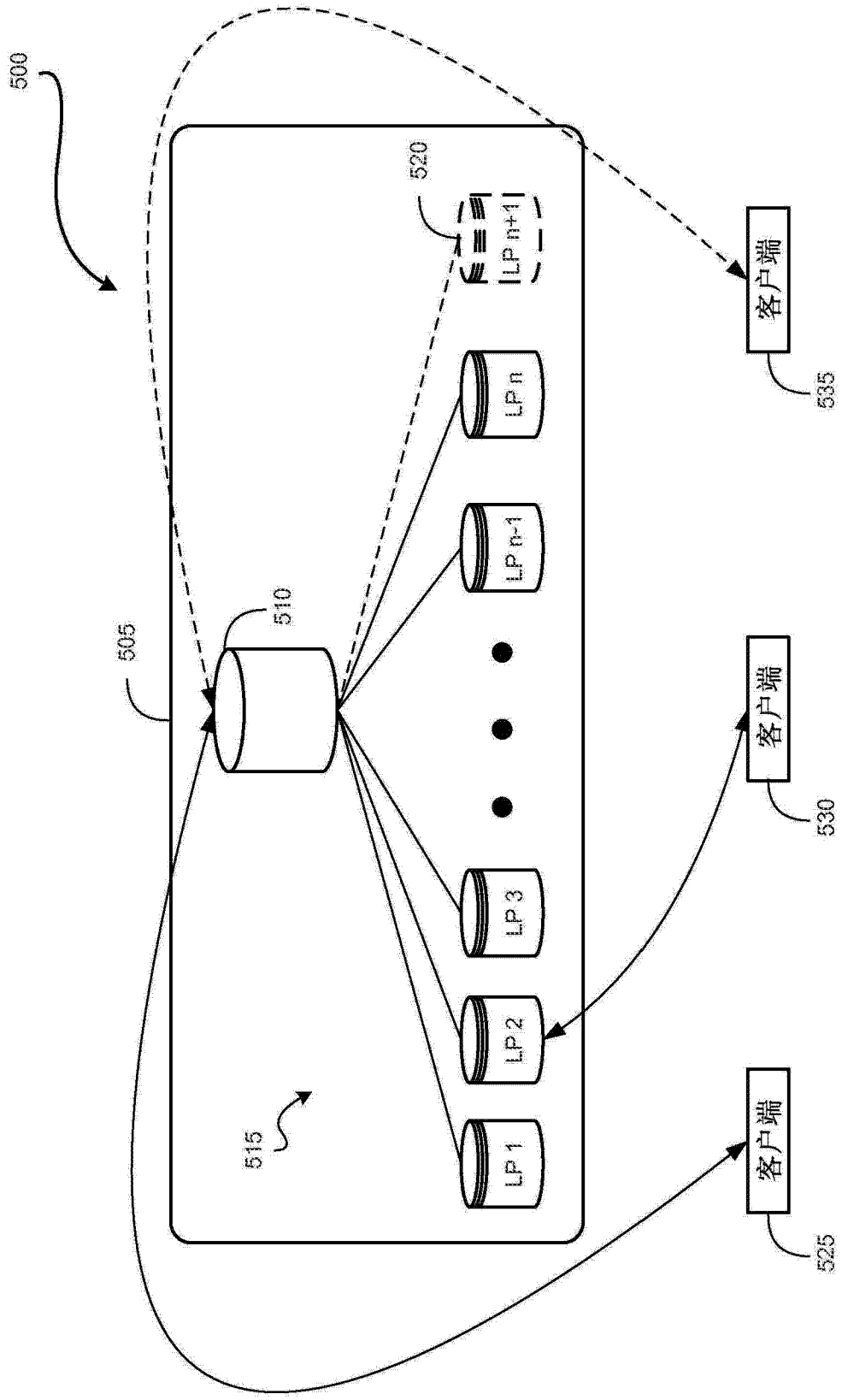

500
520
LP n+1
客户端
535
LP n
LP n-1
510
505
客户端
530
LP 3
LP 2
515
LP 1
客户端
525

图 5

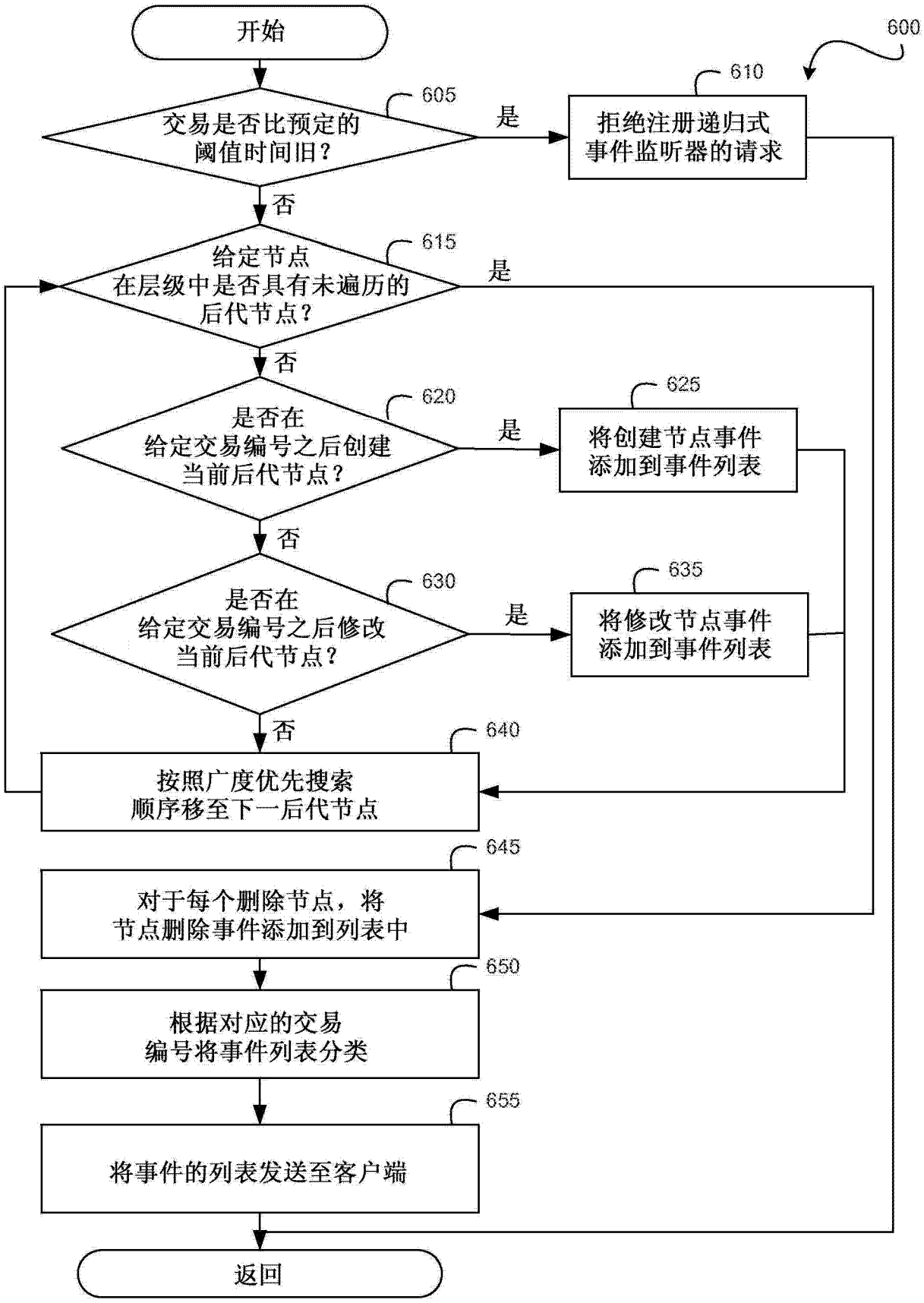

开始
600
610
605
交易是否比预定的
阈值时间旧？
是
拒绝注册递归式
事件监听器的请求
否
615
给定节点
在层级中是否具有未遍历的
后代节点？
是
否
620
625
是否在
给定交易编号之后创建
当前后代节点？
是
将创建节点事件
添加到事件列表
否
630
635
是否在
给定交易编号之后修改
当前后代节点？
是
将修改节点事件
添加到事件列表
否
640
按照广度优先搜索
顺序移至下一后代节点
645
对于每个删除节点，将
节点删除事件添加到列表中
650
根据对应的交易
编号将事件列表分类
655
将事件的列表发送至客户端
返回

图 6

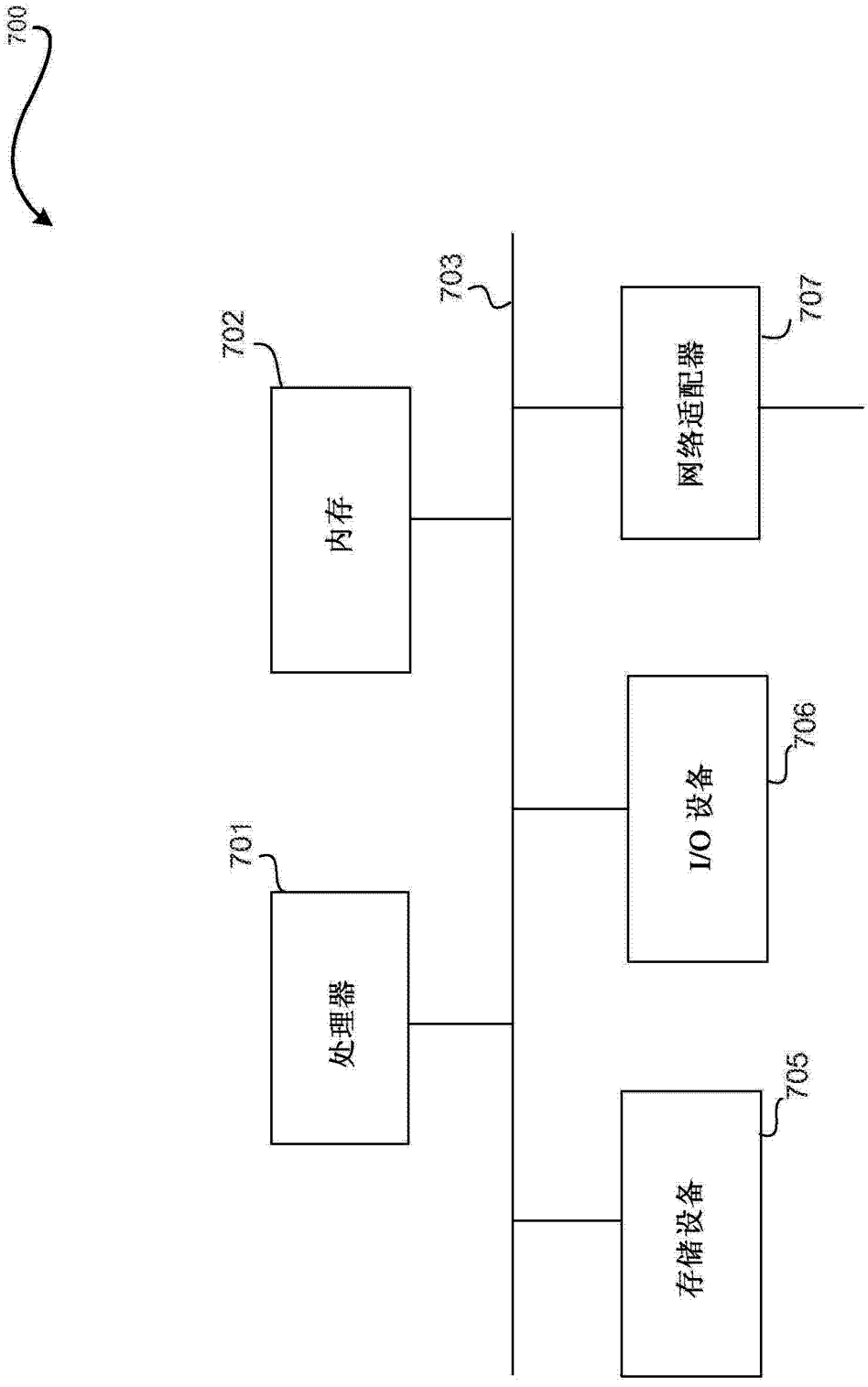


图 7