



(12) 发明专利

(10) 授权公告号 CN 108475215 B

(45) 授权公告日 2022.07.22

(21) 申请号 201780004871.2

A • 纳玛奇瓦亚姆

(22) 申请日 2017.08.03

(74) 专利代理机构 中国贸促会专利商标事务所
有限公司 11038

(65) 同一申请的已公布的文献号

专利代理人 边海梅

申请公布号 CN 108475215 A

(43) 申请公布日 2018.08.31

(51) Int.CI.

(30) 优先权数据

G06F 16/21 (2019.01)

15/227,897 2016.08.03 US

G06F 9/50 (2006.01)

(85) PCT国际申请进入国家阶段日

H04L 67/148 (2022.01)

2018.06.19

H04L 67/1001 (2022.01)

H04L 67/51 (2022.01)

(86) PCT国际申请的申请数据

(56) 对比文件

PCT/US2017/045283 2017.08.03

EP 2273367 A2, 2011.01.12

(87) PCT国际申请的公布数据

US 2003084098 A1, 2003.05.01

W02018/027026 EN 2018.02.08

IN 2011CHENP08682 A1, 2013.03.22

(73) 专利权人 甲骨文国际公司

US 2006031388 A1, 2006.02.09

地址 美国加利福尼亚

CN 101005503 A, 2007.07.25

(72) 发明人 J • 德拉瓦瑞恩 S • 弗玛

US 2014372484 A1, 2014.12.18

V • 赫格德 C • S • K • 马赫德哈拉

审查员 朱晓岗

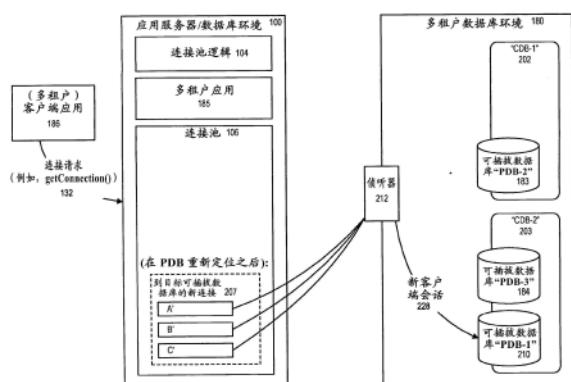
权利要求书3页 说明书7页 附图10页

(54) 发明名称

用于在多租户数据库环境中提供租户的动态重新定位的系统和方法

(57) 摘要

本文描述的是用于在多租户环境中提供对数据库的访问的系统和方法，包括使用连接池以及支持租户的动态重新定位。根据实施例，软件应用可以代表租户从连接池获得连接，该连接使得软件应用或租户能够访问数据库。重新定位处理使得与多租户或其它客户端应用相关联的租户能够在数据库环境内（例如，跨多个容器数据库）进行重新定位，其中对于客户端应用来说停机时间接近零，重新定位包括管理现有连接的释放以及新连接的迁移，而无需改变底层应用。



1. 一种用于在多租户环境中提供对数据库的访问的系统,包括使用连接池以及支持租户的动态重新定位,所述系统包括:

计算机,所述计算机包括处理器以及在所述处理器上执行的应用服务器或数据库环境中的至少一个,其中所述数据库环境包括多个容器数据库和可插拔数据库位置;

连接池,所述连接池与数据库驱动程序一起操作,并且使得软件应用能够从所述连接池中请求连接,并且使用所提供的连接来访问所述数据库环境;

其中由应用服务器或数据库环境托管的多租户应用通过使用一个或多个服务、租户到服务的映射以及数据库驱动程序,提供对数据库环境的访问,包括其中:

与所述多租户应用相关联的租户使用所述连接池来访问容器数据库的可插拔数据库,以及

每个特定租户与该特定租户在容器数据库处的自己的可插拔数据库相关联,并且经由与和该租户相关联的特定可插拔数据库相关联的数据库服务、使用由连接池提供的连接来访问该特定可插拔数据库;以及

其中响应于与数据库环境中的改变相关联的事件,所述连接池使得与客户端应用相关联的租户能够跨多个数据库位置进行重新定位,包括:

控制到原来与所述租户相关联的数据库位置的连接的释放,以及

将连接迁移到与所述租户相关联的新数据库位置,

其中所述系统还包括侦听器,所述侦听器与所述数据库位置相关联并且与所述连接池分离,并且被配置为响应于所述数据库环境中的改变,对于使用指向侦听器的连接字符串的客户端应用,将重新定向发送到提供所述连接池的应用服务器或数据库环境中的所述至少一个处的数据库驱动程序,以使得所述数据库驱动程序将新连接请求发送到所述新数据库位置,

其中所述侦听器从所述连接池和数据库驱动程序接收所述新连接请求并且将所述新连接请求转发到与所述租户相关联的所述新数据库位置,而不需要与所述租户相关联的客户端应用改变与其相关联的连接字符串。

2. 如权利要求1所述的系统,其中在释放现有连接并且将新连接从第一容器数据库处的第一可插拔数据库迁移到第二容器数据库处的新位置期间,

第二可插拔数据库在所述第二容器数据库处被打开,以及

客户端会话在所述第一可插拔数据库上被终止,并且被使得能够重新连接到与所述新位置相关联的迁移的服务。

3. 如权利要求1或2所述的系统,其中使用系统事件向所述连接池通知原来与所述租户相关联的所述数据库位置正在关闭,并且关闭相关联的连接并准备迁移。

4. 如权利要求1或2所述的系统,其中所述系统使得软件应用能够将特定标记与特定连接状态相关联。

5. 如权利要求1或2所述的系统,其中所述连接池支持多个租户,包括与每个租户相关联的不同数据库位置。

6. 一种用于在多租户环境中提供对数据库的访问的方法,包括使用连接池以及支持租户的动态重新定位,所述方法包括:

在包括处理器和在所述处理器上执行的应用服务器或数据库环境中的至少一个的计

计算机处,提供包括连接对象的连接池,并且该连接池使得软件应用能够从所述连接池中请求连接,并且使用所提供的连接来访问数据库,其中所述数据库环境包括多个容器数据库和可插拔数据库位置;

其中由应用服务器或数据库环境托管的多租户应用通过使用一个或多个服务、租户到服务的映射以及数据库驱动程序,提供对数据库环境的访问,包括其中:

与所述多租户应用相关联的租户使用所述连接池来访问容器数据库的可插拔数据库,以及

每个特定租户与该特定租户在容器数据库处的自己的可插拔数据库相关联,并且经由与和该租户相关联的特定可插拔数据库相关联的数据库服务、使用由连接池提供的连接来访问该特定可插拔数据库;以及

响应于与数据库环境中的改变相关联的事件,由所述连接池将与客户端应用相关联的租户跨多个数据库位置进行重新定位,包括:

控制到原来与所述租户相关联的数据库位置的连接的释放,

将连接迁移到与所述租户相关联的新数据库位置,以及

提供侦听器,所述侦听器与所述数据库位置相关联并且与所述连接池分离,并且被配置为响应于所述数据库环境中的改变,对于使用指向侦听器的连接字符串的客户端应用,将重新定向发送到提供所述连接池的应用服务器或数据库环境中的所述至少一个处的数据库驱动程序,以使得所述数据库驱动程序将新连接请求发送到所述新数据库位置,

其中所述侦听器从所述连接池和数据库驱动程序接收所述新连接请求并且将所述新连接请求转发到与所述租户相关联的所述新数据库位置,而不需要与所述租户相关联的客户端应用改变与其相关联的连接字符串。

7. 如权利要求6所述的方法,其中在释放现有连接并且将新连接从第一容器数据库处的第一可插拔数据库迁移到第二容器数据库处的新位置期间,

第二可插拔数据库在所述第二容器数据库处被打开,以及

客户端会话在所述第一可插拔数据库上被终止,并且被使得能够重新连接到与所述新位置相关联的迁移的服务。

8. 如权利要求6或7所述的方法,其中使用系统事件向所述连接池通知原来与所述租户相关联的数据库位置正在关闭,并且关闭相关联的连接并准备迁移。

9. 如权利要求6或7所述的方法,其中使得软件应用能够将特定标记与特定连接状态相关联。

10. 如权利要求6或7所述的方法,其中所述连接池支持多个租户,包括与每个租户相关联的不同数据库位置。

11. 一种非暂态计算机可读存储介质,包括存储在其上的指令,所述指令在由一个或多个计算机读取和执行时,使得所述一个或多个计算机执行包括以下的方法:

在包括处理器和在所述处理器上执行的应用服务器或数据库环境中的至少一个的计算机处,提供包括连接对象的连接池,并且该连接池使得软件应用能够从所述连接池中请求连接,并且使用所提供的连接来访问数据库,其中所述数据库环境包括多个容器数据库和可插拔数据库位置;

其中由应用服务器或数据库环境托管的多租户应用通过使用一个或多个服务、租户到

服务的映射以及数据库驱动程序,提供对数据库环境的访问,包括其中:

与所述多租户应用相关联的租户使用所述连接池来访问容器数据库的可插拔数据库,以及

每个特定租户与该特定租户在容器数据库处的自己的可插拔数据库相关联,并且经由与和该租户相关联的特定可插拔数据库相关联的数据库服务、使用由连接池提供的连接来访问该特定可插拔数据库;以及

响应于与数据库环境中的改变相关联的事件,由所述连接池将与客户端应用相关联的租户跨多个数据库位置进行重新定位,包括:

控制到原来与所述租户相关联的数据库位置的连接的释放,

将连接迁移到与所述租户相关联的新数据库位置,以及

提供侦听器,所述侦听器与所述数据库位置相关联并且与所述连接池分离,并且被配置为响应于所述数据库环境中的改变,对于使用指向侦听器的连接字符串的客户端应用,将重新定向发送到提供所述连接池的应用服务器或数据库环境中的所述至少一个处的数据库驱动程序,以使得所述数据库驱动程序将新连接请求发送到所述新数据库位置,

其中所述侦听器从所述连接池和数据库驱动程序接收所述新连接请求并且将所述新连接请求转发到与所述租户相关联的所述新数据库位置,而不需要与所述租户相关联的客户端应用改变与其相关联的连接字符串。

12. 如权利要求11所述的非暂态计算机可读存储介质,其中在释放现有连接并且将新连接从第一容器数据库处的第一可插拔数据库迁移到第二容器数据库处的新位置期间,

第二可插拔数据库在所述第二容器数据库处被打开,以及

客户端会话在所述第一可插拔数据库上被终止,并且被使得能够重新连接到与所述新位置相关联的迁移的服务。

13. 如权利要求11或12所述的非暂态计算机可读存储介质,其中使用系统事件向所述连接池通知原来与所述租户相关联的数据库位置正在关闭,并且关闭相关联的连接并准备迁移。

14. 如权利要求11或12所述的非暂态计算机可读存储介质,其中使得软件应用能够将特定标记与特定连接状态相关联。

15. 如权利要求11或12所述的非暂态计算机可读存储介质,其中所述连接池支持多个租户,包括与每个租户相关联的不同数据库位置。

16. 一种包括用于执行如权利要求6至10中任一项所述的方法的部件的装置。

用于在多租户数据库环境中提供租户的动态重新定位的系统 和方法

[0001] 版权声明

[0002] 本专利文献的公开内容的一部分包含受版权保护的素材。版权拥有者不反对任何人对专利文献或专利公开内容按照其在专利商标局的专利文件或记录中出现的那样进行传真复制,但是除此之外在任何情况下都保留所有版权。

[0003] 要求优先权

[0004] 本申请要求于2016年8月3日提交的申请号为15/228,897的标题为“SYSTEM AND METHOD FOR PROVIDING DYNAMIC RELOCATION OF TANANTS IN A MULTITENANT DATABASE ENVIRONMENT”的美国专利申请的优先权的权益,该申请通过引用并入本文。

技术领域

[0005] 本发明的实施例一般而言涉及软件应用服务器和数据库,并且特别地涉及用于提供对多租户环境中的数据库的访问的系统和方法,包括使用连接池以及支持租户的动态重新定位。

背景技术

[0006] 在数据库环境中,一般描述的连接池作为连接对象的高速缓存进行操作,连接对象中的每个连接对象表示可以由软件应用用来连接到数据库的连接。在运行时,应用可以从连接池中请求连接。如果连接池包括可以满足特定请求的连接,则该连接池可以将该连接返回给应用以供其使用。在一些情况下,如果未找到合适的连接,则可以创建新连接并将该新连接返回给应用。应用可以借用连接来访问数据库并执行一些工作,然后将连接返回给池,其中可以使该连接随后可用于来自相同应用或来自其它应用的后续连接请求。

发明内容

[0007] 本文描述的是用于在多租户环境中提供对数据库的访问的系统和方法,包括使用连接池以及支持租户的动态重新定位。根据实施例,软件应用可以代表租户从连接池获得连接,该连接使得软件应用或租户能够访问数据库。重新定位处理使得与多租户或其它客户端应用相关联的租户能够在数据库环境内(例如,跨多个容器数据库)进行重新定位,其中对于客户端应用来说停机时间接近零,重新定位包括管理现有连接的释放(drain)以及新连接的迁移,而无需改变底层应用。

附图说明

[0008] 图1图示了根据实施例的包括连接池的系统。

[0009] 图2进一步图示了根据实施例的包括连接池的系统,包括支持使用分片数据库。

[0010] 图3进一步图示了根据实施例的包括连接池的系统,包括支持在多租户环境中使用。

- [0011] 图4图示了根据实施例的支持在连接池环境中的租户的动态重新定位。
- [0012] 图5进一步图示了根据实施例的在连接池环境中支持租户的动态重新定位。
- [0013] 图6进一步图示了根据实施例的在连接池环境中支持租户的动态重新定位。
- [0014] 图7进一步图示了根据实施例的在连接池环境中支持租户的动态重新定位。
- [0015] 图8进一步图示了根据实施例的在连接池环境中支持租户的动态重新定位。
- [0016] 图9进一步图示了根据实施例的在连接池环境中支持租户的动态重新定位。
- [0017] 图10图示了根据实施例的在连接池环境中提供支持租户的动态重新定位的方法。

具体实施方式

[0018] 如上所述,连接池作为连接对象的高速缓存进行操作,连接对象中的每个连接对象表示可以由软件应用用来连接到数据库的连接。在运行时,应用可以从连接池中请求连接。如果连接池包括可以满足特定请求的连接,则该连接池可以将该连接返回给应用以供其使用。在一些情况下,如果未找到合适的连接,则可以创建新连接并将该新连接返回给应用。应用可以借用该连接来访问数据库并执行一些工作,然后将该连接返回给池,在池中可以使该连接随后可用于来自相同应用或来自其它应用的后续连接请求。

[0019] 创建连接对象在时间和资源方面可能是昂贵的。例如,诸如网络通信、认证、事务征用和存储器分配的任务都会影响创建特定连接对象所花费的时间和资源量。由于连接池允许重复使用这些连接对象,因此它们有助于减少必须创建各种对象的次数。

[0020] 连接池的一个示例是Oracle通用连接池(UCP),该UCP提供用于对Java数据库连接(JDBC)连接进行高度缓存的连接池。例如,连接池可以与JDBC驱动程序一起操作以创建到数据库的连接,该连接随后由池进行维护;并且可以基于发出请求的软件应用的性能和可用性要求被配置具有用于进一步优化池行为的属性。

- [0021] 连接标记(connection labeling)

- [0022] 图1图示了根据实施例的包括连接池的系统。

[0023] 如图1所示,根据实施例,包括物理计算机资源101(例如,处理器/CPU、存储器和网络组件)的应用服务器或数据库环境100(例如Oracle WebLogic服务器、Oracle融合中间件(Oracle Fusion Middleware)或其它应用服务器或数据库环境)可以包括或提供对数据库102(例如Oracle数据库或其它类型的数据库)的访问。

[0024] 如图1进一步所示,根据实施例,该系统还包括连接池逻辑104或程序代码,连接池逻辑104或程序代码在由计算机执行时,控制105连接池106中的连接对象的创建和使用,连接对象包括例如当前由软件应用使用的连接108,以及空闲连接110或当前未被正在使用的连接。

[0025] 软件应用可以在使用从连接池检索到的连接来访问或执行数据库处的工作之前初始化该连接。初始化的示例可以包括需要应用代码内的方法调用的简单状态重新初始化,或者包括需要在网络上往返的数据库操作的更复杂的初始化。后面这些类型的初始化的计算成本可能很显著。

[0026] 一些连接池(例如,UCP)允许使用连接池属性来配置其连接池,这些连接池属性具有get和set方法,并且通过启用池的数据源实例而成为可用的。这些get和set方法提供了以编程方式配置池的简便方法。如果未设置池属性,则连接池使用默认的属性值。

[0027] 根据实施例,标记连接允许客户端软件应用将任意名称/值对附接到连接。然后,应用可以从连接池中请求具有期望标记的连接。通过将特定标记与特定连接状态相关联,应用可以潜在地从池中检索已经初始化的连接,并且避免了重新初始化的时间和成本。连接标记不会对用户定义的键或值强加任何含义;任何用户定义的键和值的含义仅仅由应用定义。

[0028] 例如,如图1所示,根据实施例,连接池可以包括当前由软件应用使用的多个连接,这里标示为连接A 112和B 114。每个连接可以被标记,例如,连接A被标记(为蓝色)并且连接B被标记(为绿色)。这些标记/颜色是为了图示的目的而提供,并且如上所述可以是由客户端应用附接到连接的任意名称/值对。根据各种实施例,可以使用不同类型的标记来区分不同的连接类型;并且不同的应用可以将不同的标记/颜色附接到特定的连接类型。

[0029] 如图1进一步所示,根据实施例,连接池还可以包括空闲的或者当前未被软件应用正在使用的多个连接,这里标示为连接C 116、D 118、E 120、F 122、G 124和N 126。每个空闲连接可以被类似地标记,在该图示中被标记为(蓝色)或(绿色),并且再次地,这些标记/颜色是为了图示的目的而提供的。

[0030] 如图1进一步所示,根据实施例,如果软件应用130希望使用特定类型的连接(例如(红色)连接)对数据库做出请求,则应用可以做出“getConnection(red)”请求132。作为响应,连接池逻辑将创建新的(红色)连接,这里标示为X 134(红色);或者将现有空闲连接从(蓝色或绿色)重新调整用途为(红色),这里标示为E 135(红色)。

[0031] 分片数据库(sharded database)

[0032] 根据实施例,分片是使用跨多个独立物理数据库的数据的水平划分的数据库缩放技术。存储在每个物理数据库中的数据的部分被称为碎片(shard)。从软件客户端应用的角度来看,所有物理数据库的集合都表现为单个逻辑数据库。

[0033] 根据实施例,系统可以包括支持使用具有分片数据库的连接池。碎片引导器或侦听器提供软件客户端应用对数据库碎片的访问。连接池(例如,UCP)和数据库驱动程序(例如,JDBC驱动程序)可以被配置为允许客户端应用在连接检验期间或者在以后的时间提供碎片键;识别由客户端应用指定的碎片键;以及启用由客户端应用到特定碎片或块的连接。该方法使得能够高效地重用连接资源,并且能够更快地访问适当的碎片。

[0034] 图2进一步图示了根据实施例的包括连接池的系统,包括支持使用分片数据库。

[0035] 根据实施例,可以使用碎片键(SHARD_KEY)将数据库表划分为,例如确定在特定碎片内每一行被存储在哪里的一个或多个列。碎片键可以在连接串或描述中被提供为连接数据(CONNECT_DATA)的特性。碎片键的示例可以包括数据库中的VARCHAR2、CHAR、DATE(日期)、NUMBER(数值)或TIMESTAMP(时间戳)。根据实施例,分片数据库还可以接受没有碎片键或碎片组键的连接。

[0036] 根据实施例,为了减少重新分片对系统性能和数据可用性的影响,每个碎片可以被细分成更小的片或块。每个块充当可以从一个碎片移动到另一个碎片的重新分片单元。通过向碎片键映射添加间接层,块也简化了路由。

[0037] 例如,每个块可以自动地与碎片键值的范围相关联。用户提供的碎片键可以映射到特定的块,并且该块可以映射到特定的碎片。如果数据库操作尝试在特定碎片上不存在的块上操作,则将会出现错误。当使用碎片组时,每个碎片组是具有碎片组标识符的特定值

的那些块的集合。

[0038] 碎片感知的客户端应用可以与分片数据库配置一起工作,包括连接到其中数据基于一个或多个分片方法被划分的一个或多个数据库碎片的能力。每次需要数据库操作时,客户端应用可以确定该客户端应用需要连接到的碎片。

[0039] 根据实施例,分片方法可以用于将碎片键值映射到各个碎片。可以支持不同的分片方法,例如:基于散列的分片,其中向每个块分配一定范围的散列值,使得在建立数据库连接时,系统将散列函数应用到分片键的给定值,并计算对应的散列值,该散列值随后基于该值所属的范围被映射到块;基于范围的分片,其中一定范围的碎片键值被直接分配给各个碎片;以及基于列表的分片,其中每个碎片与碎片键值列表相关联。

[0040] 如图2所示,根据实施例,分片数据库140可以包括第一数据库区域A(这里标示为“DB东”,DBE)141,DBE 141包括具有被存储为块A1、A2、...、An的碎片A的分片数据库实例“DBE-1”142;以及具有被存储为块B1、B2、...、Bn的碎片B的“DBE-2”143。

[0041] 如图2中进一步所示,根据实施例,第二数据库区域B(这里标示为“DB西”,DBW)144包括具有被存储为块C1、C2、...、Cn的碎片C的分片数据库实例“DBW-1”145;以及具有被存储为块D1、D2、...、Dn的碎片D的“DBW-2”146。

[0042] 根据实施例,每个数据库区域或分片数据库实例组可以与碎片引导器或侦听器(例如,Oracle全球服务管理器(GSM)侦听器或另一种类型的侦听器)相关联。例如,如图2所示,碎片引导器或侦听器147可以与第一数据库区域A相关联;并且另一个碎片引导器或侦听器148可以与第二数据库区域B相关联。系统可以包括维护碎片拓扑层154的数据库驱动程序(例如,JDBC驱动程序)152,其在一段时间内学习碎片键范围并且将碎片键范围高速缓存到分片数据库中每个碎片的位置。

[0043] 根据实施例,客户端应用可以在连接请求162期间向连接池提供一个或多个碎片键;并且基于该一个或多个碎片键以及由碎片拓扑层提供的信息,连接池可以将连接请求路由到正确或适当的碎片。

[0044] 根据实施例,连接池还可以通过其碎片键来标识到特定碎片或块的连接,并且当从特定客户端应用接收到对相同碎片键的请求时允许重用连接。

[0045] 例如,如图2所示,根据实施例,到特定块(例如,块A1)的连接可以用来连接174到该块。如果池中不存在到特定碎片或块的可用连接,则系统可以尝试将现有的可用连接重新调整用途给另一个碎片或块,并重用该连接。可以使数据库中跨碎片和块的数据分布对客户端应用是透明的,这也最小化了客户端上块的重新分片的影响。

[0046] 当碎片感知的客户端应用向连接池提供与连接请求相关联的一个或多个碎片键时;则如果连接池或数据库驱动程序已经具有碎片键的映射,则可以将连接请求直接转发到适当的碎片和块,在这个示例中,转发到块C2。

[0047] 当碎片感知的客户端应用不提供与连接请求相关联的碎片键时;或者如果连接池或数据库驱动程序不具有所提供的碎片键的映射;则可以将连接请求转发到适当的碎片引导器或侦听器。

[0048] 多租户环境

[0049] 根据实施例,系统可以包括使用连接标记支持基于云的环境或多租户环境。例如,多租户云环境可以包括应用服务器或数据库环境,该应用服务器或数据库环境包括或提供

在基于云的环境中对数据库的访问以供多个租户或租户应用使用。

[0050] 图3进一步图示了根据实施例的包括连接池的系统,包括支持在多租户环境中使用。

[0051] 类似于上述环境,可以由租户经由云或其它网络访问的软件应用可以在使用连接之前对从连接池检索到的连接进行初始化。

[0052] 如上所述,初始化的示例可以包括需要应用代码内的方法调用的简单状态重新初始化,或者包括需要在网络上往返的数据库操作的更复杂的初始化。

[0053] 还如上所述,标记连接允许应用将任意名称/值对附接到连接,使得应用随后可以从连接池中请求具有期望标记的连接,包括从池中检索已经初始化的连接的能力,并且避免了重新初始化的时间和成本。

[0054] 如图3所示,根据实施例,多租户数据库环境180可以包括例如容器数据库 (CDB) 181和一个或多个可插拔数据库 (PDB),这里示为“PDB-1”182、“PDB-2”183和“PDB-3”184。

[0055] 根据实施例,每个PDB可以与多租户应用的租户(这里标示为“租户-1”、“租户-2”和“租户-3”)相关联,多租户应用由应用服务器或数据库环境185托管或者被提供为外部客户端应用186,并且通过使用一个或多个Oracle真正应用集群 (Real Application Cluster, RAC) 实例186、188(在这个示例中包括“RAC-Instance-1”和“RAC-Instance-2”);一个或多个服务(在这个示例中包括“服务-1”、“服务-2”和“服务-3”)以及租户到服务的映射190来提供对数据库环境的访问。

[0056] 在图3所示的示例中,由租户用来访问数据库环境的应用可以做出与该租户的数据源192、194、196相关联的连接请求,并且系统可以在需要的情况下切换服务198以利用到现有RAC实例或PDB的连接。

[0057] 服务器侧连接池

[0058] 根据实施例,系统可以利用服务器侧连接池标签 (tagging) 特征,诸如例如由 Oracle数据库驻留连接池 (DRCP) 提供的特征。服务器侧连接池标签特征允许用户应用或客户端基于使用该数据库环境所理解的单个标签来选择性地获取到数据库环境的连接。

[0059] 根据实施例,每个连接仅关联一个标签。数据库服务器不将标签值传送给用户应用或客户端,而是传送标签匹配(例如,作为布尔 (Boolean) 值)。

[0060] 池中租户的动态重新定位

[0061] 根据实施例,系统可以包括支持租户的动态重新定位。软件应用可以代表租户从连接池获得使得软件应用或租户能够访问数据库的连接。重新定位处理使得与多租户或其它客户端应用相关联的租户能够在数据库环境内(例如,跨多个容器数据库)进行重新定位,其中对于客户端应用来说停机时间接近零,重新定位包括管理现有连接的释放以及新连接的迁移,而无需改变底层应用。

[0062] 图4-图9图示了根据实施例的在连接池环境中支持租户的动态重新定位。

[0063] 如图4所示,根据实施例,数据库(例如,容器数据库(例如,“CDB-1”202)或另一种类型的数据库)支持使用多个连接204。

[0064] 与由应用服务器或数据库环境托管的或被提供为外部客户端应用的多租户或其它客户端应用相关联的租户可以使用连接池来访问数据库,包括在适当的情况下经由数据库服务访问容器数据库的可插拔数据库。

[0065] 例如,根据实施例,每个特定租户可以与该特定租户在容器数据库处的自己的特定可插入数据库相关联,并且可以使用由连接池提供的连接经由与特定可插拔数据库关联的数据库服务来访问(例如205)与该租户关联的特定可插拔数据库。

[0066] 根据实施例,如果数据库环境改变,例如,第二容器数据库(例如,“CDB-2”203)被添加到系统,或者响应于正在托管连接池的应用服务器从数据库环境接收服务关闭事件206,系统可以将新连接207提供给新数据库位置以供特定租户使用。

[0067] 例如,根据实施例,系统可以发起可插拔数据库的迁移以供租户使用,包括释放与原始可插拔数据库位置及其相关联的数据库服务相关联的连接(例如,与“CDB-1”202中的“PDB-1”182相关联的那些连接);以及将这些连接208的可用性迁移或以其它方式重新定位到新的可插拔数据库位置和相关联的数据库服务(例如,这里在“CDB-2”203中标示为“PDB-1”210)。

[0068] 这使得连接池能够通过以对客户端或租户应用透明的方式释放与租户的原始位置相关联的现有连接并且创建指向租户的新位置的新连接来支持接近零停机时间的租户重新定位。

[0069] 例如,在多租户环境中,系统支持将与特定租户相关联的可插拔数据库从第一Oracle真正应用集群(RAC)数据库移动到第二RAC数据库;或从第一容器数据库移动到第二容器数据库。

[0070] 但是,这些可插拔数据库通常作为不同/单独的数据库进行操作,这可能导致连接丢失。

[0071] 为了解决这个问题,根据实施例,在当前正在使用指向原始容器数据库(例如,“CDB-1”)的侦听器212的连接字符串的应用的情况下,侦听器可以被配置为将连接请求重新定向到新位置或容器数据库(例如,“CDB-2”)。这允许侦听器将重新定向发送到应用服务器处的数据库驱动程序,这进而导致数据库驱动程序将新连接请求发送到新容器数据库。

[0072] 此外,现有的连接请求必须从原始容器数据库中释放出来。但是,池可能还不知道关于新容器数据库的存在,因为它被认为是不同的数据库。

[0073] 为了解决这个问题,根据实施例,可以使用系统事件通知(例如,Oracle通知服务事件)向连接池通知可插拔数据库正在关闭,并且关闭该数据库相关联的连接并准备迁移到与新位置相关联的新数据库服务。

[0074] 通常,存在短的时间段,在此期间新数据库位置不会立即可用于支持新连接。在此期间,现有连接将被关闭,并且连接池在接收到新请求之前不会创建新连接。这可能导致轻微的系统停机,例如,由于需要更新重做日志(redo log),包括停止重做日志以将事实来源切换到新位置。

[0075] 例如,在图4所示的示例中,其中期望将可插拔数据库(例如,“PDB-1”)从第一容器数据库(例如,“CDB-1”)迁移到第二容器数据库(例如“CDB-2”);然后,根据实施例,在可插拔数据库的重新定位中涉及的处理包括:

[0076] 1.发起可插拔数据库的重新定位。例如,如图5所示,服务器可以通过运行“更改可插拔数据库重新定位(alter pluggable database relocate)”命令来发起可插拔数据库的重新定位,这将影响在原始可插拔数据库上运行的那些会话220。

[0077] 2.在新位置处打开可插拔数据库,然后终止原始实例容器数据库上的所有客户端

会话。例如,如图6所示,系统可以通过在容器数据库实例“CDB-2”中打开可插拔数据库“PDB-1”来响应“更改可插拔数据库重新定位”命令,然后终止原始容器数据库实例“CDB-1”上的所有客户端会话。此后,系统将关闭“CDB-1”上的可插拔数据库“PDB-1”,并清空该数据库的缓冲区高速缓存。

[0078] 3.使得客户端能够重新连接到新数据库位置。例如,如图7所示,客户端然后将需要自己重新连接到(现在被迁移的)服务226。连接池以对应用透明的方式实现这一点,例如,如图8所示,包括通过在从服务器接收到服务关闭事件时释放现有连接,并重新创建到迁移的可插拔数据库的新连接。

[0079] 4.将连接请求转发到新位置。例如,如图9所示,在服务器侧,一旦迁移完成,侦听器就将把新连接请求228从连接池转发到新的目标容器数据库(例如“CDB-2”)。应用不需要改变其连接串,这使得重新定位处理对应用透明。

[0080] 动态重新定位处理

[0081] 图10图示了根据实施例的在连接池环境中提供支持租户的动态重新定位的方法。

[0082] 如图10所示,根据实施例,在步骤231处,在应用服务器或数据库环境处,提供控制连接池中的连接对象的创建和使用的连接池逻辑或程序代码,其中软件应用可以从连接池中请求连接,并使用所提供的连接来访问数据库。

[0083] 如图10所示,根据实施例,在步骤233处,接收用以将与租户相关联的可插拔数据库从第一容器数据库实例迁移到第二容器数据库实例处的新位置的指令。

[0084] 在步骤235处,服务器发起可插拔数据库的重新定位,这影响在可插拔数据库上运行的那些会话。

[0085] 在步骤237处,系统通过在新位置处打开可插拔数据库,然后终止第一容器数据库实例上的所有客户端会话来作出响应。

[0086] 在步骤239处,使得客户端能够重新连接到与新位置相关联的(迁移的)服务。

[0087] 在步骤241处,在服务器侧,一旦迁移完成,侦听器就将新连接请求从连接池转发到(新)容器数据库位置。

[0088] 本发明的实施例可以使用一个或多个常规的通用或专用数字计算机、计算设备、机器或微处理器(包括根据本公开的教导进行编程的一个或多个处理器、存储器和/或计算机可读存储介质)来方便地实现。如对软件领域技术人员将显而易见的,基于本公开的教导,适当的软件编码可以由熟练的程序员容易地准备。

[0089] 在一些实施例中,本发明包括计算机程序产品,计算机程序产品是其上/其中存储有指令的非暂态存储介质或计算机可读介质,指令可以用来对计算机进行编程以执行本发明的任何处理。存储介质的示例可以包括但不限于任何类型的盘(包括软盘、光盘、DVD、CD-ROM、微驱动器和磁光盘)、ROM、RAM、EPROM、EEPROM、DRAM、VRAM、闪存设备,磁卡或光卡、纳米系统(包括分子存储器IC),或适于存储指令和/或数据的任何类型的介质或设备。

[0090] 本发明的实施例的前述描述是为了说明和描述的目的而提供的。它不旨在是详尽的或者将本发明限于所公开的精确形式。许多修改和变化对于本领域技术人员将是显而易见的。修改和变化包括所公开特征的任何相关组合。实施例的选择和描述是为了最好地解释本发明的原理及其实际应用,由此使本领域的其他技术人员能够理解本发明的各种实施例以及适于预期的特定用途的各种修改。

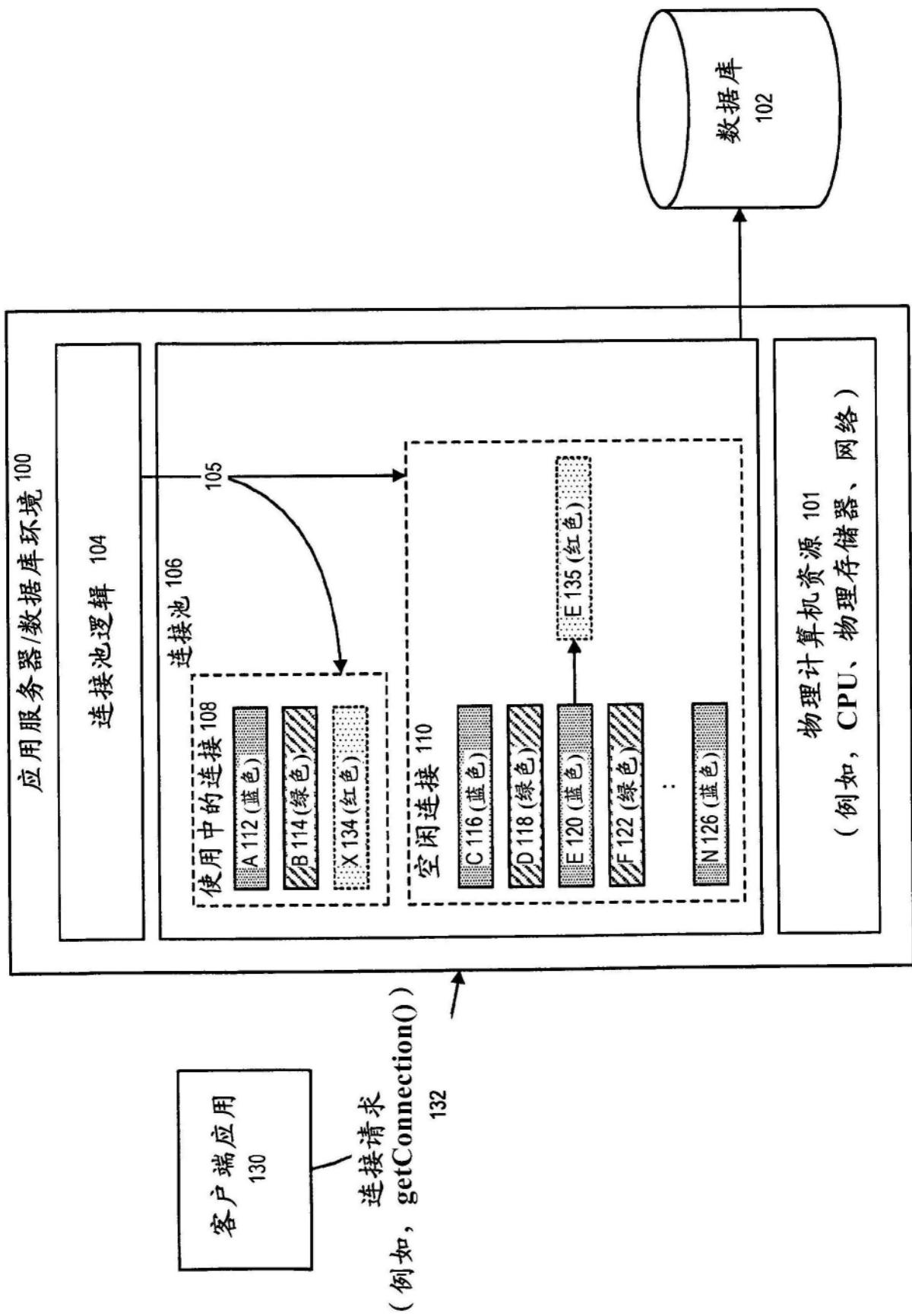


图1

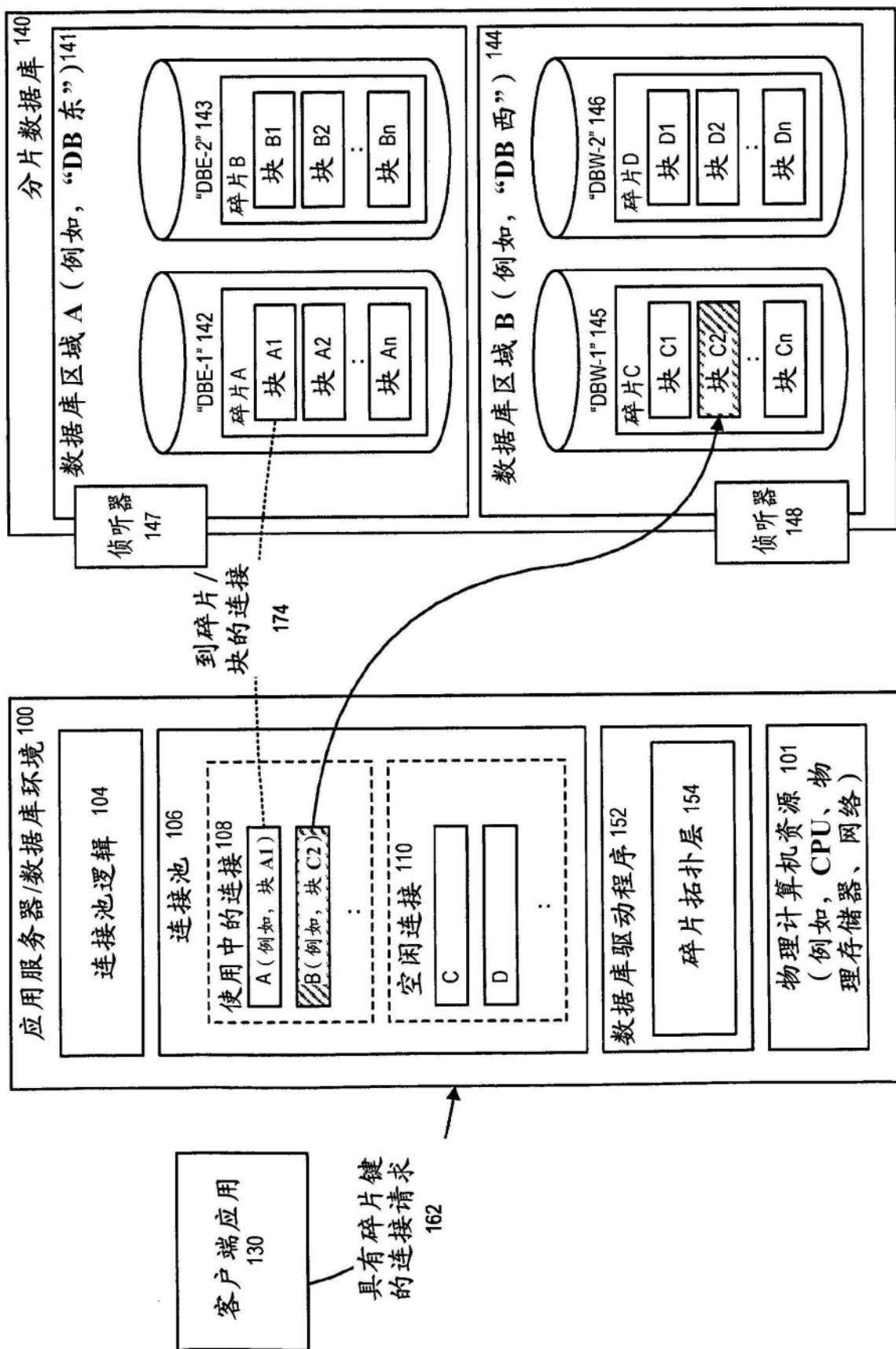


图2

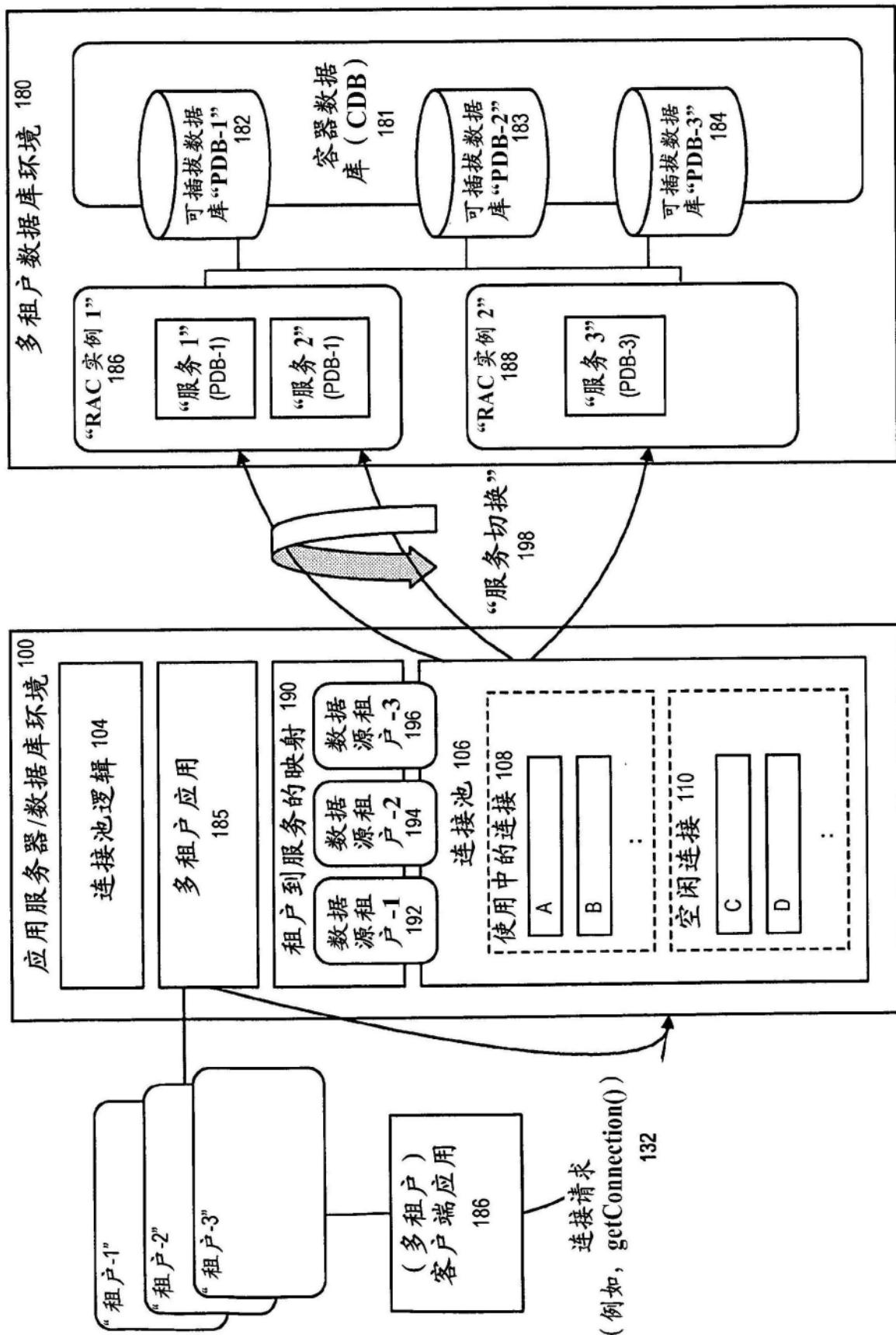


图3

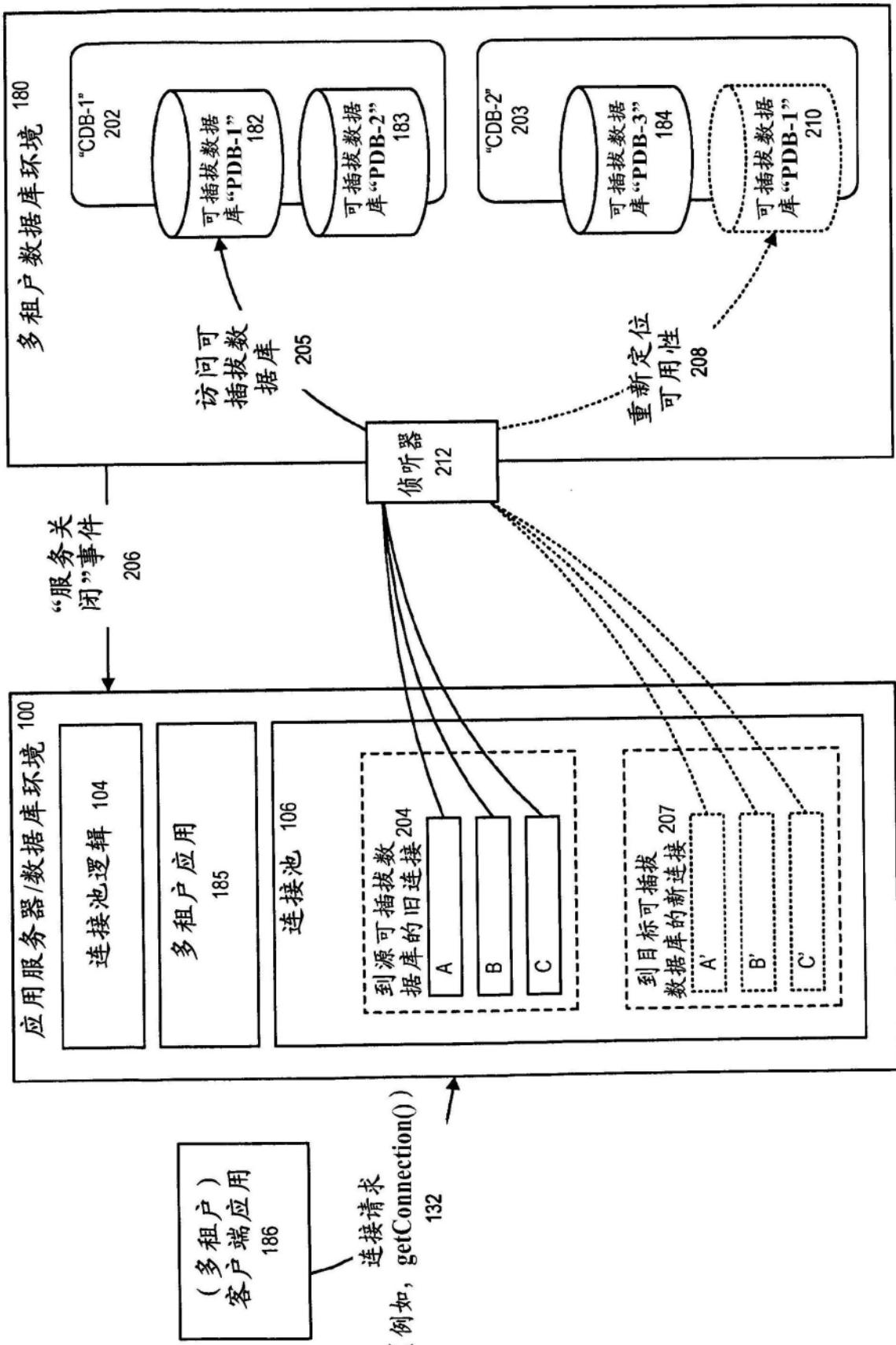
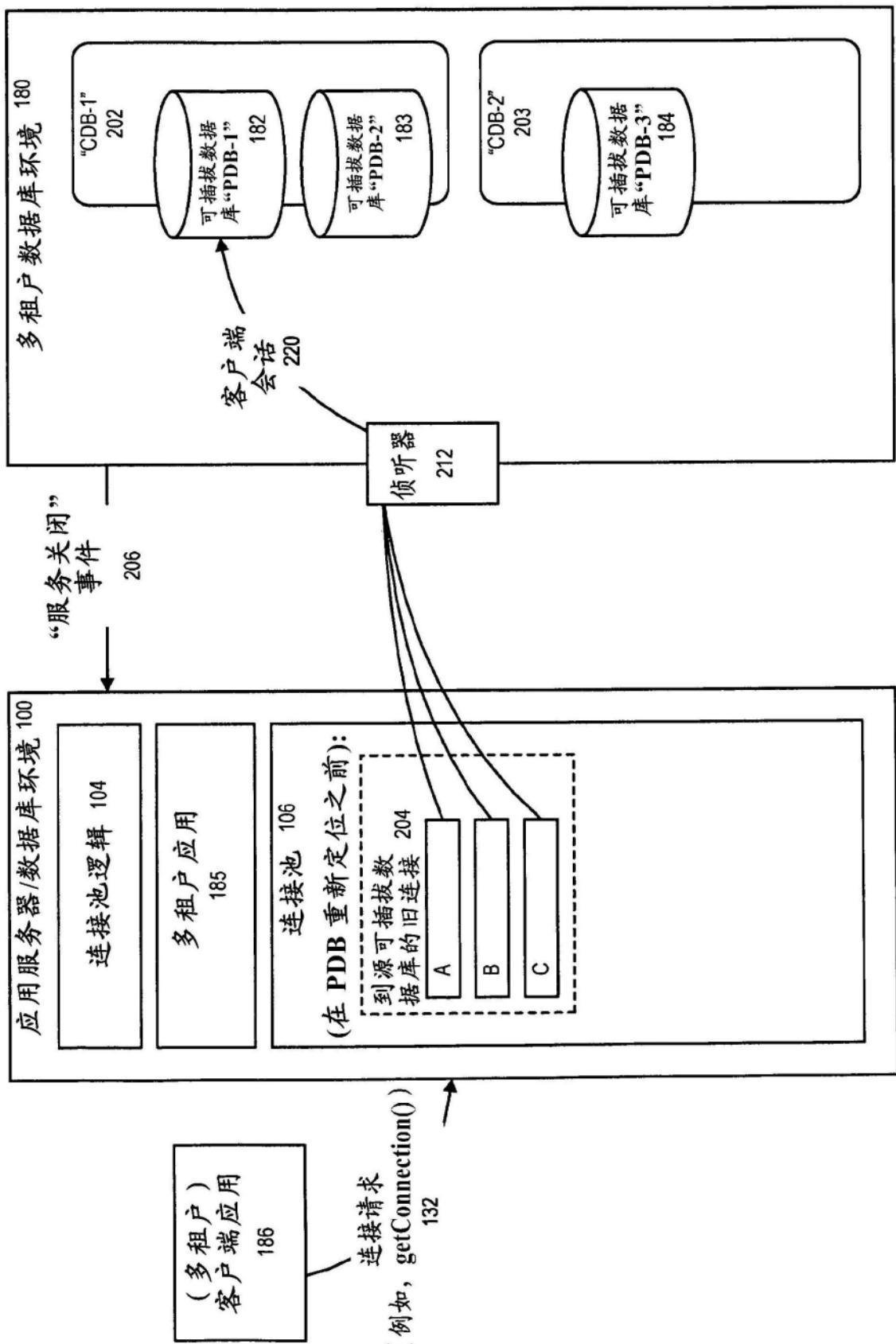


图4



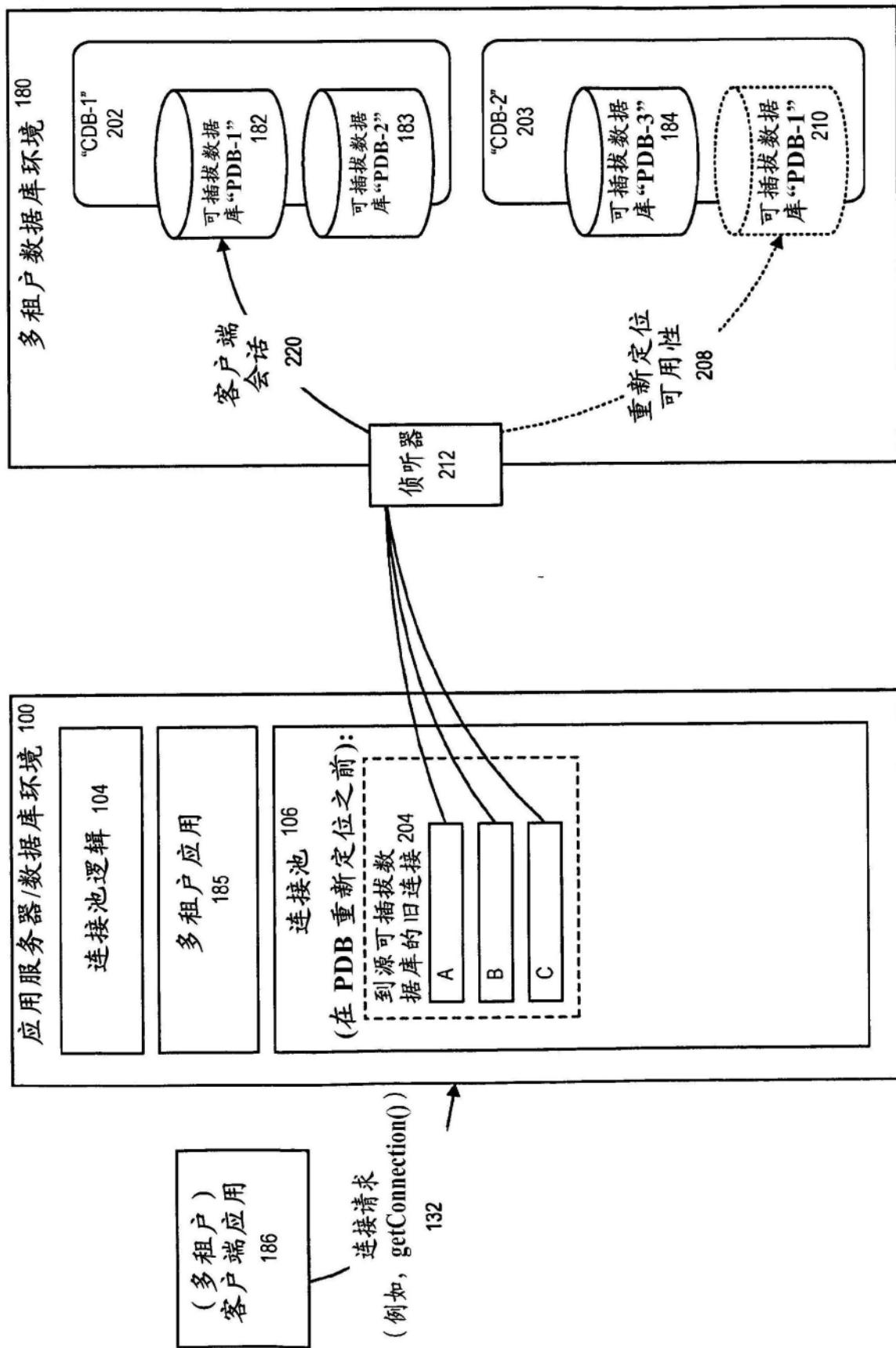


图6

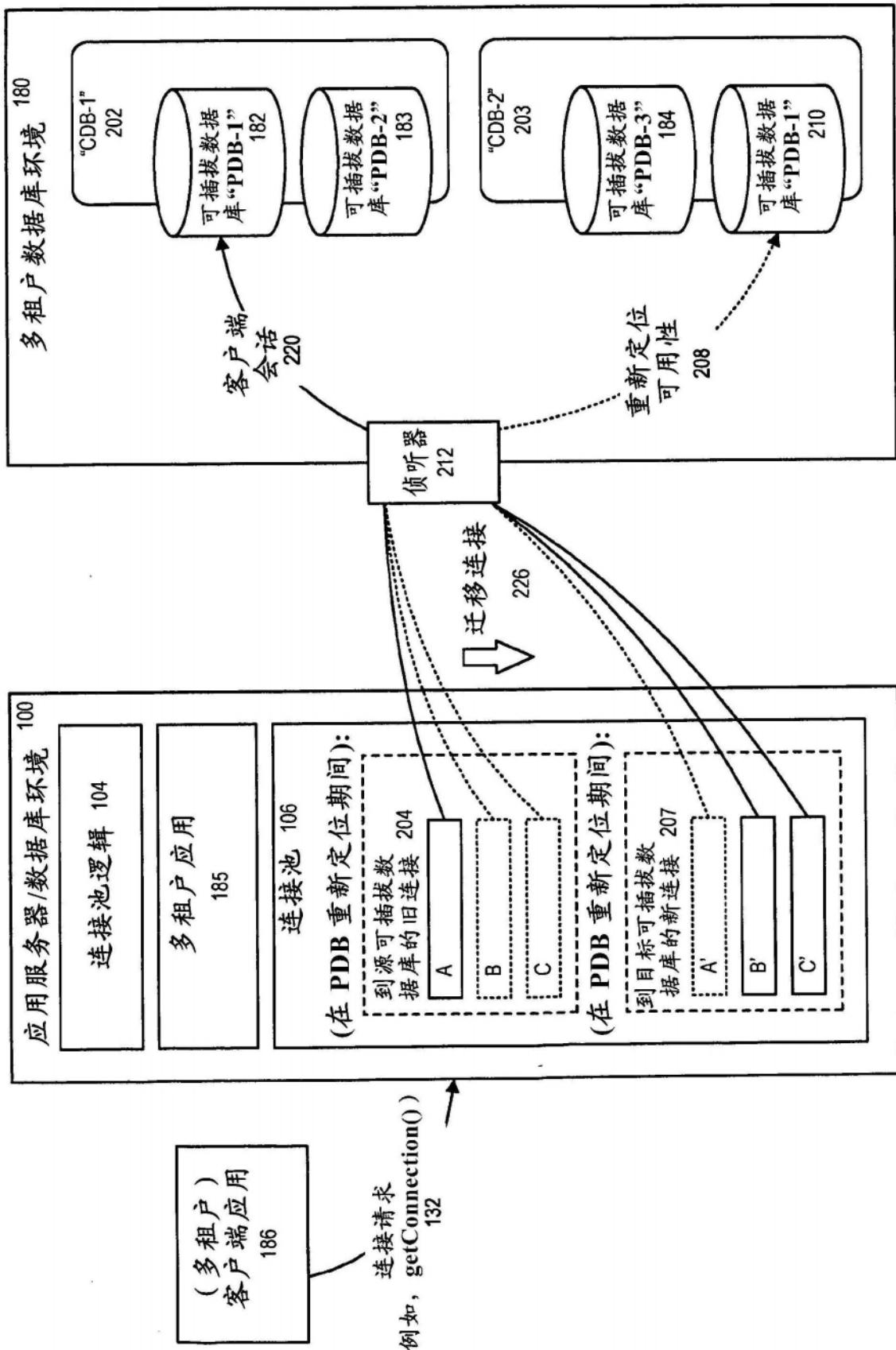
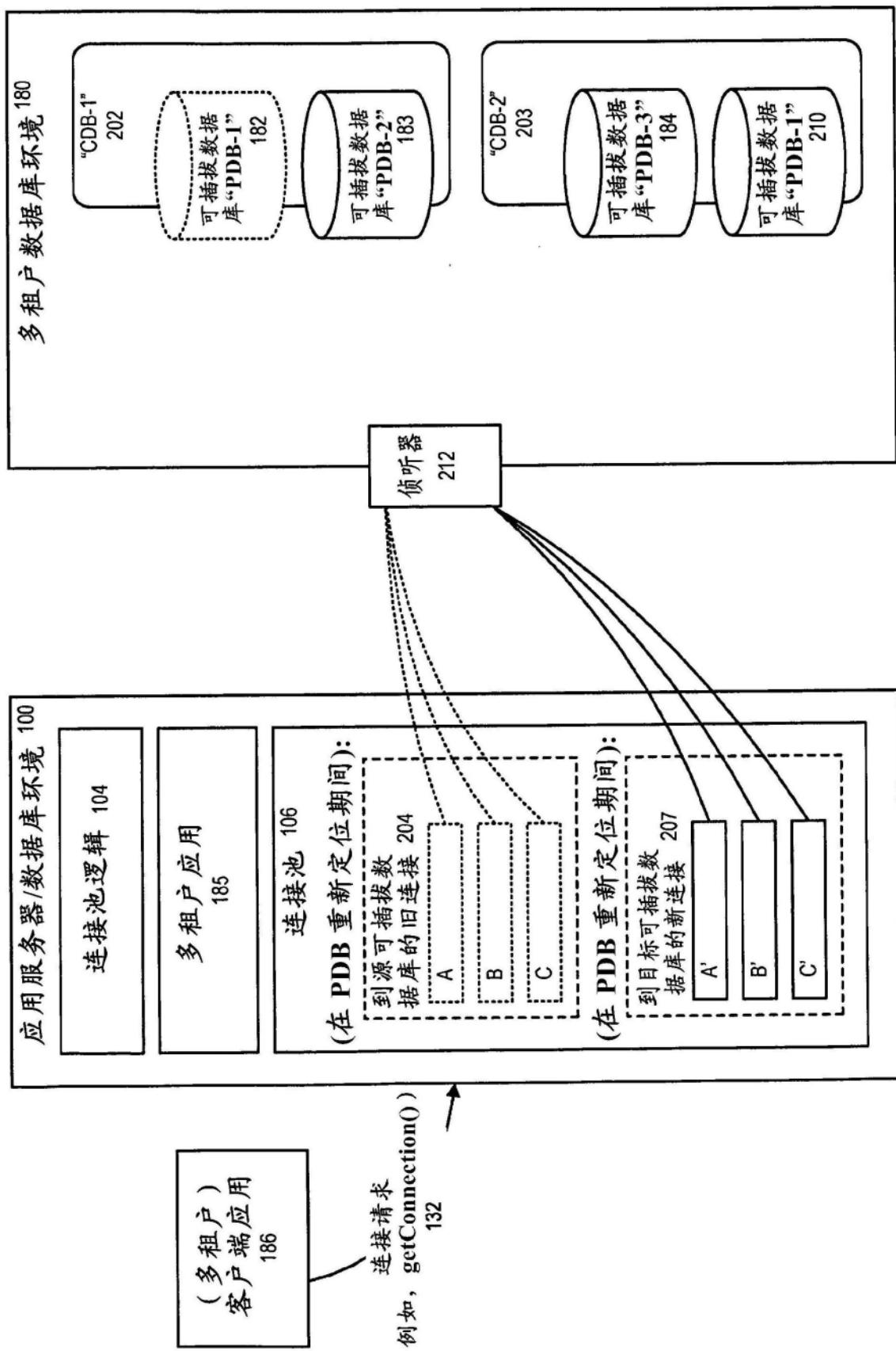
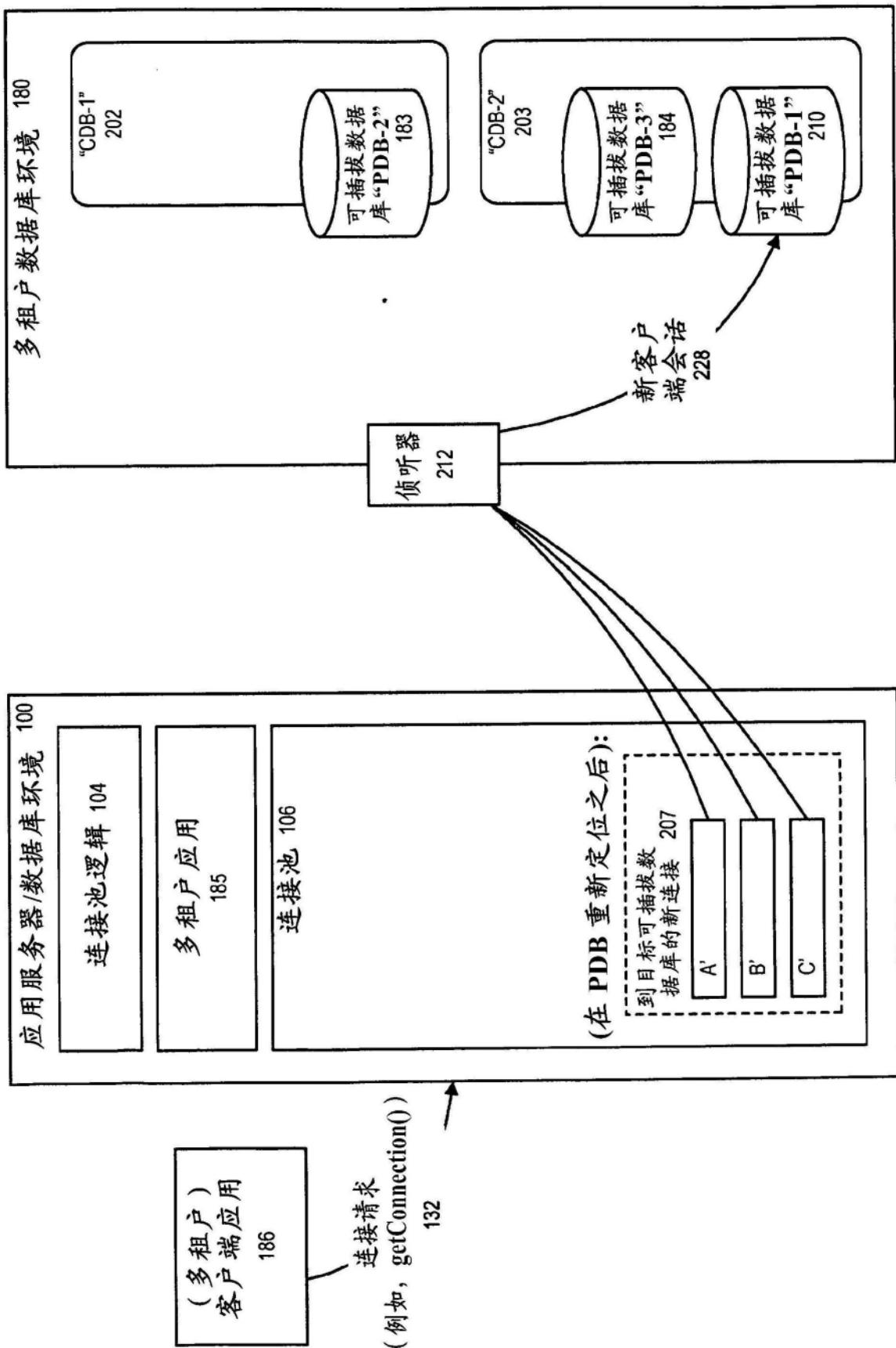


图 7





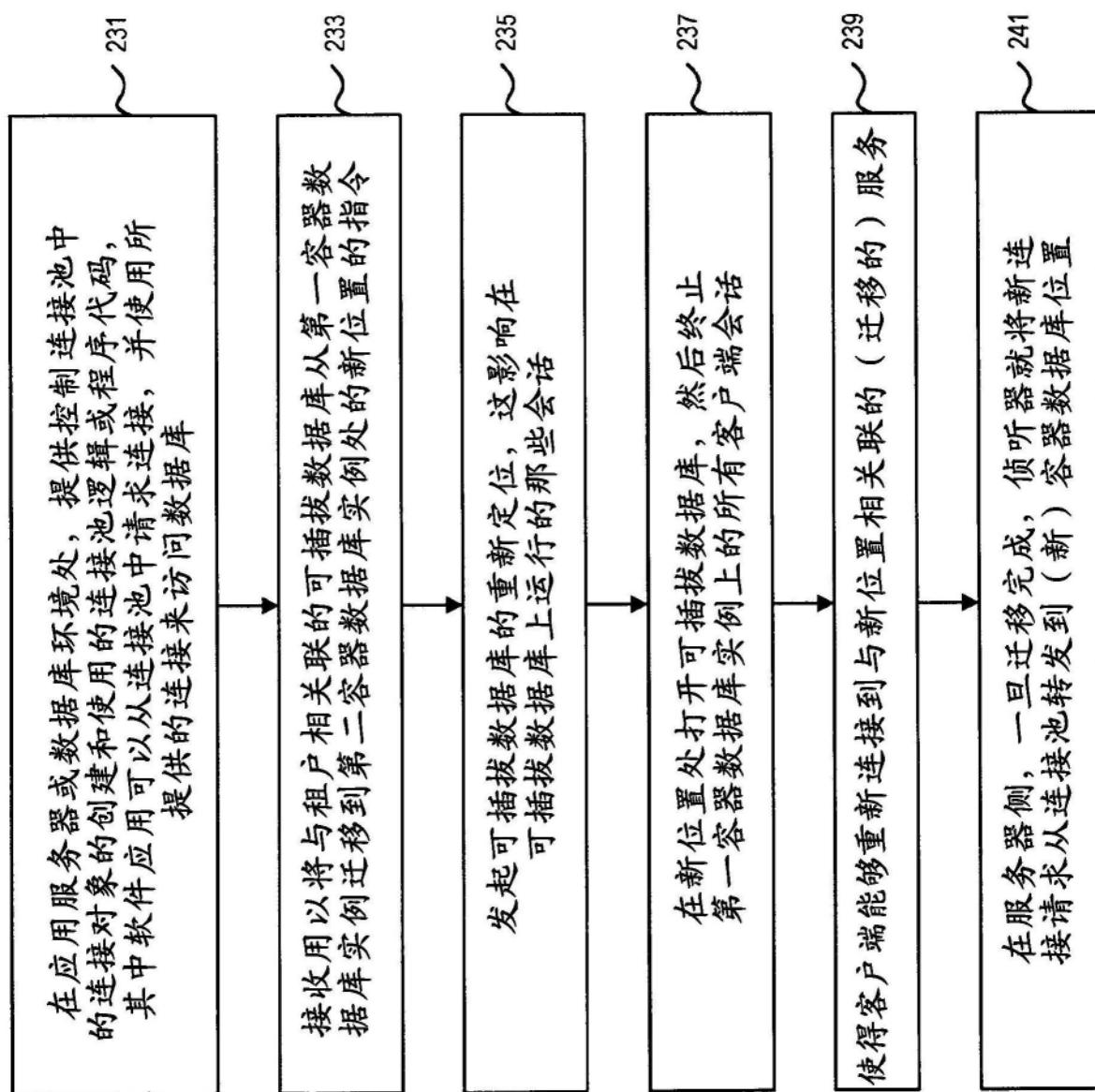


图10