



(12)发明专利

(10)授权公告号 CN 106462470 B

(45)授权公告日 2020.01.21

(21)申请号 201580033969.1

(22)申请日 2015.06.23

(65)同一申请的已公布的文献号
申请公布号 CN 106462470 A

(43)申请公布日 2017.02.22

(30)优先权数据
62/016,032 2014.06.23 US
62/054,906 2014.09.24 US

(85)PCT国际申请进入国家阶段日
2016.12.23

(86)PCT国际申请的申请数据
PCT/US2015/037262 2015.06.23

(87)PCT国际申请的公布数据
W02015/200372 EN 2015.12.30

(73)专利权人 甲骨文国际公司

地址 美国加利福尼亚

(72)发明人 C·帕特尔 J·迪坡 T·阿加沃
N·伊斯拉姆 R·莫达尼

(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 李晓芳

(51)Int.Cl.
G06F 9/50(2006.01)

(56)对比文件
US 2013304788 A1,2013.11.14,
CN 101639835 A,2010.02.03,
US 2013326506 A1,2013.12.05,

审查员 刘褚燚

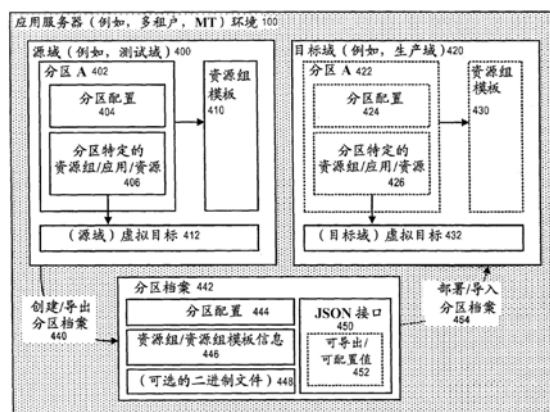
权利要求书3页 说明书14页 附图8页

(54)发明名称

用于在多租户应用服务器环境中的可移植分区的系统和方法

(57)摘要

根据实施例,本文描述了用于在多租户应用服务器环境中提供可移植分区的系统和方法。根据实施例,系统使得分区能够从第一域或源域(例如,测试域)导出,并且被导入到另一第二域或目标域(例如,生产域)中。在分区档案的创建或导出期间,创建分区档案,其包括分区配置、资源组或资源组模板信息以及可选地包括应用二进制文件。分区档案还可以包括允许访问与分区相关联的可导出或可配置值的接口。在将分区档案部署或导入到目标域期间,包括与分区相关联的可导出或可配置值在内的分区档案中的信息被用来配置分区以供在目标域中使用。



1. 一种用于在应用服务器环境中支持可移植分区的使用的系统,包括:
一个或多个计算机,包括应用服务器环境,所述应用服务器环境具有:
用于软件应用的执行的域配置,所述域配置限定所述应用服务器环境内的域,以及
一个或多个分区和资源组,
其中每个分区与分区标识符和分区配置相关联,并且提供所述域的管理和运行时细分并且能够包括一个或多个资源组,以及
其中每个资源组是能够引用资源组模板的可部署应用或资源的集合,
其中多个分区能够各自使用分区特定的配置信息来运行相同的资源组;以及
其中所述系统使得分区能够从第一应用服务器域导出,并且被导入到第二应用服务器域中,包括其中
创建分区档案,所述分区档案与所述分区相关联并且包括所述分区的分区特定的配置、资源组或资源组模板信息以及可选地包括应用二进制文件,并且
所述分区档案用于将所述分区导入到第二应用服务器域中。
2. 根据权利要求1所述的系统,其中所述应用服务器或云环境包括Java EE应用服务器。
3. 根据权利要求1或2所述的系统,其中所述分区档案包括允许访问与所述分区相关联的可导出或可配置值的接口。
4. 根据权利要求3所述的系统,其中在将所述分区档案部署或导入到目标域期间,包括与所述分区相关联的所述可导出或可配置值在内的所述分区档案中的信息被用来配置所述分区以供在所述目标域中使用,其中与所述分区相关联的所述可导出或可配置值包括从管理员或从另一源接收的任何经修改的可配置值。
5. 根据权利要求1至2和4中的任何一项权利要求所述的系统,其中所述系统在云环境内提供,以支持在所述云环境内操作的多个租户,包括所述系统能够可选地将一个或多个分区与租户相关联,以供所述租户使用。
6. 一种用于在应用服务器环境中支持可移植分区的使用的方法,包括:
在一个或多个计算机处,提供应用服务器环境,所述应用服务器环境具有:
用于软件应用的执行的域配置,所述域配置限定所述应用服务器环境内的域,以及
一个或多个分区和资源组,
其中每个分区与分区标识符和分区配置相关联,并且提供所述域的管理和运行时细分并且能够包括资源组,以及
其中每个资源组是能够引用资源组模板的可部署应用或资源的集合,
其中多个分区能够各自使用分区特定的配置信息来运行相同的资源组;以及
其中所述方法使得分区能够从第一应用服务器域导出,并且被导入到第二应用服务器域中,包括其中
创建分区档案,所述分区档案与所述分区相关联并且包括所述分区的分区特定的配置、资源组或资源组模板信息以及可选地包括应用二进制文件,并且
所述分区档案用于将所述分区导入到第二应用服务器域中。
7. 根据权利要求6所述的方法,其中所述应用服务器或云环境包括Java EE应用服务器。

8. 根据权利要求6或7所述的方法,其中所述分区档案包括允许访问与所述分区相关联的可导出或可配置值的接口。

9. 根据权利要求8所述的方法,其中在将所述分区档案部署或导入到目标域期间,包括与所述分区相关联的所述可导出或可配置值在内的所述分区档案中的信息被用来配置所述分区以供在所述目标域中使用,其中与所述分区相关联的所述可导出或可配置值包括从管理员或从另一源接收的任何经修改的可配置值。

10. 根据权利要求6至7和9中任何一项权利要求所述的方法,其中所述方法在云环境内执行,以支持在所述云环境内操作的多个租户,包括所述方法能够可选地将一个或多个分区与租户相关联,以供所述租户使用。

11. 一种计算设备,包括:

一个或多个处理器;

计算机可读存储介质,所述计算机可读存储介质包括以机器可读格式存储在其上的程序指令,当所述程序指令被所述一个或多个处理器执行时,使得所述一个或多个处理器执行如权利要求6至10中的任何一项权利要求所述的方法。

12. 一种包括用于执行如权利要求6至10中的任何一项权利要求所述的方法的部件的装置。

13. 一种非暂态计算机可读存储介质,包括存储在其上的指令,指令当被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步骤,包括:

提供应用服务器环境,所述应用服务器环境具有:

用于软件应用的执行的域配置,所述域配置限定所述应用服务器环境内的域,以及一个或多个分区和资源组,

其中每个分区与分区标识符和分区配置相关联,并且提供所述域的管理和运行时细分并且能够包括资源组,以及

其中每个资源组是能够引用资源组模板的可部署应用或资源的集合,

其中多个分区能够各自使用分区特定的配置信息来运行相同的资源组;以及

其中所述指令当被所述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使得分区能够从第一应用服务器域导出,并且被导入到第二应用服务器域中,包括其中

创建分区档案,所述分区档案与所述分区相关联并且包括所述分区的分区特定的配置、资源组或资源组模板信息以及可选地包括应用二进制文件,并且

所述分区档案用于将所述分区导入到第二应用服务器域中。

14. 根据权利要求13所述的非暂态计算机可读存储介质,其中所述应用服务器或云环境包括Java EE应用服务器。

15. 根据权利要求13或14所述的非暂态计算机可读存储介质,其中所述分区档案包括允许访问与所述分区相关联的可导出或可配置值的接口。

16. 根据权利要求15所述的非暂态计算机可读存储介质,其中在将所述分区档案部署或导入到目标域期间,包括与所述分区相关联的所述可导出或可配置值在内的所述分区档案中的信息被用来配置所述分区以供在所述目标域中使用,其中与所述分区相关联的所述可导出或可配置值包括从管理员或从另一源接收的任何经修改的可配置值。

17. 根据权利要求13至14和16中的任何一项权利要求所述的非暂态计算机可读存储介

质,其中所述步骤在云环境内执行,以支持在所述云环境内操作的多个租户,包括一个或多个分区能够可选地与租户相关联,以供所述租户使用。

用于在多租户应用服务器环境中的可移植分区的系统和方法

[0001] 版权声明

[0002] 本专利文档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档或专利公开内容按照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。

技术领域

[0003] 本发明的实施例一般涉及应用服务器和云环境,并且更具体地涉及用于在多租户应用服务器环境中的可移植分区的系统和方法。

背景技术

[0004] 应用服务器一般提供软件应用可以在其中被部署并运行的受管理环境。基于云的环境允许应用在云提供的分布式资源内运行并且利用云提供的分布式资源。这样的环境可以支持大量的用户或租户,这些用户或租户中的一些可以具有特定于该用户或租户的具体需求。这些是本发明的实施例可以在其中被使用的环境类型的一些示例。

发明内容

[0005] 根据实施例,本文描述了用于在多租户应用服务器环境中提供可移植分区的系统和方法。根据实施例,系统使得分区能够从第一域或源域(例如,测试域)导出,并且被导入到另一第二域或目标域(例如,生产域)中。在分区档案(archive)的创建或导出期间,创建分区档案,其包括分区配置、资源组或资源组模板信息以及可选地包括应用二进制文件。分区档案还可以包括允许访问与分区相关联的可导出或可配置值的接口。在将分区档案部署或导入到目标域期间,包括与分区相关联的可导出或可配置值在内的分区档案中的信息被用来配置分区以在目标域中使用。

附图说明

[0006] 图1示出了根据实施例的、用于在应用服务器、云或其它环境中支持多租赁的系统。

[0007] 图2进一步示出了根据实施例的、用于在应用服务器、云或其它环境中支持多租赁的系统。

[0008] 图3进一步示出了根据实施例的、用于在应用服务器、云或其它环境中支持多租赁的系统。

[0009] 图4示出了根据实施例的、用于与示例性多租户环境一起使用的域配置。

[0010] 图5进一步示出了根据实施例的示例性多租户环境。

[0011] 图6示出了根据实施例的、在应用服务器、云或其它环境中可移植分区的使用。

[0012] 图7进一步示出了根据实施例的、在应用服务器、云或其它环境中可移植分区的使用。

[0013] 图8示出了根据实施例的、在应用服务器、云或其它环境中使用可移植分区的方法。

具体实施方式

[0014] 根据实施例,本文描述了用于在多租户应用服务器环境中提供可移植分区的系统和方法。根据实施例,系统使得分区能够从第一域或源域(例如,测试域)导出,并且被导入到另一第二域或目标域(例如,生产域)中。在分区档案的创建或导出期间,创建分区档案,其包括分区配置、资源组或资源组模板信息以及可选地包括应用二进制文件。分区档案还可以包括允许访问与分区相关联的可导出或可配置值的接口。在将分区档案部署或导入到目标域期间,包括与分区相关联的可导出或可配置值在内的分区档案中的信息被用来配置分区以在目标域中使用。

[0015] 应用服务器(例如,多租户,MT)环境

[0016] 图1示出了根据实施例的用于在应用服务器、云或其它环境中支持多租赁的系统。

[0017] 如图1中所示,根据实施例,实现软件应用的部署和执行的应用服务器(例如,多租户,MT)环境100或其它计算环境可被配置为包括在运行时被用来定义应用服务器域的域102配置并且根据该域102配置来操作。

[0018] 根据实施例,应用服务器可以包括被定义以用于在运行时使用的一个或多个分区104。每个分区可以与全局唯一的分区标识符(ID)和分区配置相关联,并且还可以包括一个或多个资源组124,连同对资源组模板的引用126和/或分区特定的应用或资源128。域级资源组、应用和/或资源140也可以在域级被定义,可选地具有对资源组模板的引用。

[0019] 每个资源组模板160可以定义一个或多个应用A 162、B 164、资源A 166、B 168和/或其它可部署的应用或资源170,并且可以由资源组来引用。例如,如图1中所示,分区104中的资源组124可以引用190资源组模板160。

[0020] 一般而言,系统管理员可以定义分区、域级资源组和资源组模板以及安全领域;而分区管理员可以例如通过创建分区级资源组、将应用部署到分区或者引用用于分区的具体领域来定义其自己的分区的方面。

[0021] 图2进一步示出了根据实施例的、用于在应用服务器、云或其它环境中支持多租赁的系统。

[0022] 如图2中所示,根据实施例,分区202可以包括例如资源组205,资源组205包括对资源组模板210的引用206、虚拟目标(例如,虚拟主机)信息207以及可插拔数据库(PDB)信息208。资源组模板(例如,210)可以定义例如多个应用A 211和B 212,连同诸如Java消息服务器(JMS)服务器213、存储转发(SAF)代理215、邮件会话部件216或Java数据库连接(JDBC)资源217之类的资源。

[0023] 图2中所示的资源组模板通过示例的方式被提供;根据其它实施例,可以提供不同类型的资源组模板和元素。

[0024] 根据实施例,当分区(例如,202)内的资源组引用220特定的资源组模板(例如,210)时,与特定分区相关联的信息可以与所引用的资源组模板结合使用,以指示分区特定信息230,例如分区特定的PDB信息。然后,分区特定信息可以由应用服务器用来配置资源(例如PDB资源)以供分区使用。例如,与分区202关联的分区特定的PDB信息可以由应用服务

器用来利用适当的PDB 238配置232容器数据库 (CDB) 236,以供该分区使用。

[0025] 类似地,根据实施例,与特定分区相关联的虚拟目标信息可被用来定义239分区特定的虚拟目标240(例如,baylandurgentcare.com),以供该分区使用,然后可以使该分区特定的虚拟目标240经由统一资源定位符 (URL) (例如,http://baylandurgentcare.com)可访问。

[0026] 图3进一步示出了根据实施例的用于在应用服务器、云或其它环境中支持多租赁的系统。

[0027] 根据实施例,诸如config.xml配置文件之类的系统配置被用来定义分区,该系统配置包括用于与该分区相关联的资源组的配置元素,和/或其它分区特性。可以使用特性名称/值对来为每个分区地指定值。

[0028] 根据实施例,多个分区可以在可以提供对CDB 243的访问并且经由web层244可访问的受管理的服务器/集群242或者类似环境内执行。这允许例如域或分区与 (CDB的) PDB中的一个或多个PDB相关联。

[0029] 根据实施例,多个分区中的每个分区 (在该示例中为分区A 250和分区B 260) 可被配置为包括与该分区相关联的多个资源。例如,分区A可被配置为包括资源组251,资源组251包含应用A1 252、应用A2 254和JMS A 256,连同与PDB A 259相关联的数据源A 257,其中该分区可经由虚拟目标A 258访问。类似地,分区B 260可被配置为包括资源组261,资源组261包含应用B1 262、应用B2 264和JMS B 266,连同与PDB B 269相关联的数据源B 267,其中该分区可经由虚拟目标B 268访问。

[0030] 虽然上面的示例中的几个示例示出了CDB和PDB的使用,但是根据其它实施例,可以支持其它类型的多租户或非多租户数据库,其中可以例如通过模式的使用或不同数据库的使用来为每个分区提供特定的配置。

[0031] 资源

[0032] 根据实施例,资源是可被部署到环境的域的系统资源、应用或者其它资源或对象。例如,根据实施例,资源可以是可被部署到服务器、集群或其它应用服务器目标的应用、JMS、JDBC、JavaMail、WLDF、数据源或者其它系统资源或其它类型的对象。

[0033] 分区

[0034] 根据实施例,分区是可以与分区标识符 (ID) 和配置相关联、并且可以通过资源组和资源组模板的使用包含应用和/或参考域范围的资源的域的运行时和管理细分或切片。

[0035] 一般而言,分区可以包含其自己的应用、经由资源组模板参考域范围的应用,并且具有其自己的配置。可分区的实体可以包括资源,例如JMS、JDBC、JavaMail、WLDF资源,以及其它部件,诸如JNDI命名空间、网络业务、工作管理器以及安全策略和领域。在多租户环境的上下文中,系统可被配置为提供对于与租户相关联的分区的管理和运行时方面的租户访问。

[0036] 根据实施例,分区内的每个资源组可以可选地引用资源组模板。分区可以具有多个资源组,并且这些资源组中的每个资源组可以引用资源组模板。每个分区可以定义用于在该分区的资源组所引用的资源组模板中未指定的配置数据的特性。这使得分区能够充当在资源组模板中定义的可部署资源到用于与该分区一起使用的具体值的绑定。在一些情况下,分区可以重写 (override) 由资源组模板指定的配置信息。

[0037] 根据实施例,例如由config.xml配置文件定义的分区配置可以包括多个配置元素,例如:“partition(分区)”,其包含定义分区的属性和子元素;“resource-group(资源组)”,其包含被部署到分区的应用和资源;“resource-group-template(资源组模板)”,其包含由那个模板定义的应用和资源;“jdbc-system-resource-override(JDBC系统资源重写)”,其包含数据库特定的服务名称、用户名和密码;以及“partition-properties(分区特性)”,其包含可用于资源组模板中的宏替换的特性键值。

[0038] 在启动时,系统可以使用由配置文件提供的信息以从资源组模板为每个资源生成分区特定的配置元素。

[0039] 资源组

[0040] 根据实施例,资源组是可以在域或分区级被定义并且可以引用资源组模板的可部署资源的命名的、完全限定的集合。资源组中的资源被认为是完全限定的,是因为管理员已提供了启动或连接到那些资源所需的信息中的所有信息,例如用于连接到数据源的凭据或者用于应用的定向信息。

[0041] 系统管理员可以在域级或者在分区级声明资源组。在域级,资源组提供了分组相关资源的方便的方式。系统可以与未分组的资源相同地管理在域级资源组中声明的资源,以使得资源可以在系统启动期间被启动,并且在系统关闭期间被停止。管理员还可以单独地停止、启动或移除组中的资源,并且可以通过对组的操作来隐式地对组中的所有资源执行动作。例如,停止资源组停止该组中尚未停止的资源中的所有资源;启动资源组启动该组中尚未起动的任何资源;而移除资源组移除该组中包含的资源中的所有资源。

[0042] 在分区级,受任何安全限制的影响,系统或分区管理员可以在分区中配置零个或多个资源组。例如,在SaaS用例中,各种分区级资源组可以参考域级资源组模板;而在PaaS用例中,不参考资源组模板的分区级资源组可以被创建,但是替代地表示将仅在该分区内可用的应用及其相关资源。

[0043] 根据实施例,资源分组可被用来将应用和它们使用的资源分组在一起作为域内的不同管理单元。例如,在下面描述的医疗记录(MedRec)应用中,资源分组定义MedRec应用及其资源。多个分区可以运行相同的MedRec资源组,每个分区利用分区特定的配置信息,以使得作为每个MedRec实例的一部分的应用特定于每个分区。

[0044] 资源组模板

[0045] 根据实施例,资源组模板是在域级定义的、可以从资源组引用的可部署资源的集合,并且激活其资源所需的信息中的一些信息可以不被存储为模板本身的一部分,以使得其支持分区级配置的规范。域可以包含任何数量的资源组模板,这些资源组模板中的每个资源组模板可以包括例如一个或多个相关的Java应用和这些应用所依赖的资源。关于这些资源的信息中的一些信息可以是跨所有分区相同的,而其它信息可以依分区而不同。并非所有配置都需要在域级指定——分区级配置可以替代地通过宏或特性名称/值对的使用在资源组模板中指定。

[0046] 根据实施例,特定的资源组模板可以由一个或多个资源组引用。一般而言,在任何给定的分区内,资源组模板可以一次由一个资源组引用,即,不由同一分区内的多个资源组同时引用;然而,它可以由不同分区中的另一资源组同时引用。包含资源组的对象(例如域或分区)可以使用特性名称/值指派来设置资源组模板中任何标记(token)的值。当系统使

用引用的资源组激活资源组模板时,它可以用在资源组的包含对象中设置的值替换这些标记。在一些情况下,系统还可以使用静态配置的资源组模板和分区来为每个分区/模板组合生成运行时配置。

[0047] 例如,在SaaS用例中,系统可以多次激活相同的应用和资源,包括一次为了将使用它们的每个分区的激活。当管理员定义资源组模板时,他们可以使用标记来表示将在其它地方提供的信息。例如,在连接到CRM相关的数据资源中使用的用户名可以在资源组模板中被指示为\\${CRMDDataUsername}。

[0048] 租户

[0049] 根据实施例,在诸如多租户(MT)应用服务器环境之类的多租户环境中,租户是可以由一个或多个分区和/或一个或多个租户感知的应用表示或以其它方式与一个或多个分区和/或一个或多个租户感知的应用相关联的实体。

[0050] 例如,租户可以表示诸如不同的外部公司或特定企业内的不同部门(例如,HR部门和财务部门)之类的不同的用户组织,这些用户组织中的每个用户组织可以与不同的分区相关联。租户全局唯一身份(租户ID)是特定用户在特定时刻与特定租户的关联。系统可以从用户身份例如通过参考用户身份仓库(store)来导出特定用户属于哪个租户。用户身份使系统能够实施用户被授权执行的那些动作,包括但不限于用户可以属于哪个租户。

[0051] 根据实施例,系统实现将不同租户的管理和运行时彼此隔离。例如,租户可以配置他们的应用的一些行为,以及他们可以访问的资源。系统可以确保特定的租户不能管理属于另一租户的工件(artifact);并且在运行时代表特定租户工作的应用仅参考与那个租户相关联的资源,而不参考与其他租户相关联的资源。

[0052] 根据实施例,非租户感知的应用是不包含显示处理租户的逻辑的应用,以使得不管哪个用户提交了该应用正在对其进行响应的请求,该应用使用的任何资源都可以是可访问的。相比之下,租户感知的应用包括显式处理租户的逻辑。例如,基于用户的身份,应用可以导出用户所属的租户并且使用该信息来访问租户特定的资源。

[0053] 根据实施例,系统使用户能够部署被显式写为租户感知的的应用,以使得应用开发者可以获得当前租户的租户ID。然后,租户感知的应用可以使用租户ID来处理正在使用应用的单个实例的多个租户。

[0054] 例如,支持单个医生的办公室或医院的MedRec应用可以暴露给两个不同的分区或租户(例如,湾地紧急护理(Bayland Urgent Care)租户和山谷健康(Valley Health)租户),这些不同的分区或租户中的每个能够访问诸如分开的PDB之类的分开的租户特定的资源,而无需改变底层应用代码。

[0055] 示例性域配置和多租户环境

[0056] 根据实施例,应用可以在域级被部署到资源组模板,或者被部署到范围为分区或范围为域的资源组。可以使用每应用或每分区指定的部署计划重写应用配置。部署计划也可以指定为资源组的一部分。

[0057] 图4示出了根据实施例的、用于与示例性多租户环境一起使用的域配置。

[0058] 根据实施例,当系统启动分区时,它根据所提供的配置创建到各自的数据库实例的虚拟目标(例如,虚拟主机)和连接池,这包括为每个分区创建一个虚拟目标和连接池。

[0059] 通常,每个资源组模板可以包括一个或多个相关的应用和那些应用所依赖的资

源。通过提供资源组模板中的可部署资源到与分区相关联的具体值的绑定,每个分区可以提供在它所参考的资源组模板中没有指定的配置数据;在一些情况下,这包括重写由资源组模板指定的某些配置信息。这使得系统能够使用每个分区已定义的特性值为每个分区以不同方式激活由资源组模板表示的应用。

[0060] 在一些实例中,分区可以包含不参考资源组模板或直接定义它们自己的分区范围的可部署资源的资源组。在分区内定义的应用和数据源一般仅对于该分区是可用的。资源可以被部署以使得可以使用分区:<分区名称>/<资源JNDI名称>或者域:<资源JNDI名称>从跨分区来访问它们。

[0061] 例如,MedRec应用可以包括多个Java应用、数据源、JMS服务器和邮件会话。为了为多个租户运行MedRec应用,系统管理员可以定义单个MedRec资源组模板286,在该模板中声明那些可部署资源。

[0062] 与域级的可部署资源相比,在资源组模板中声明的可部署资源可能未在模板中完全配置,或者不能按原样被激活,因为它们缺少一些配置信息。

[0063] 例如,MedRec资源组模板可以声明由应用使用的数据源,但是它可以不指定用于连接到数据库的URL。与不同租户相关联的分区(例如分区BUC-A 290(湾地紧急护理,BUC)和分区VH-A 292(山谷健康,VH))可以通过各自包括引用296、297MedRec资源组模板的MedRec资源组293、294来引用一个或多个资源组模板。然后,该引用可被用来创建302、306用于每个租户的虚拟目标/虚拟主机,包括与BUC-A分区相关联的虚拟主机baylandurgentcare.com 304,以供湾地紧急护理租户使用;以及与VH-A分区相关联的虚拟主机valleyhealth.com 308,以供山谷健康租户使用。

[0064] 图5进一步示出了根据实施例的示例性多租户环境。如图5中所示,并且继续其中两个分区引用MedRec资源组模板的来自上文的示例,根据实施例,服务器端小程序(servlet)引擎310可被用来支持多个租户环境,在这个示例中是湾地紧急护理医师(Bayland Urgent Care Physician)租户环境320和山谷健康医师(Valley Health Physician)租户环境330。

[0065] 根据实施例,每个分区321、331可以定义在其上接受用于该租户环境的传入业务的不同虚拟目标,以及用于连接到该分区及其资源324、334的不同URL 322、332,在这个示例中分别包括湾地紧急护理数据库或山谷健康数据库。数据库实例可以使用兼容模式,因为相同的应用代码将对这两个数据库都执行。当系统启动分区时,它可以创建到各自的数据库实例的虚拟目标和连接池。

[0066] 可移植分区

[0067] 根据实施例,系统使得分区能够从第一域或源域(例如,测试域)导出,并且被导入到另一第二域或目标域(例如,生产域)中。在分区档案的创建或导出期间,创建分区档案,该分区档案包括分区配置、资源组或资源组模板信息以及可选地包括应用二进制文件。分区档案还可以包括允许访问与分区相关联的可导出或可配置值的接口。在将分区档案部署或导入到目标域期间,包括与分区相关联的可导出或可配置值在内的分区档案中的信息被用来配置分区以在目标域中使用。

[0068] 根据各种实施例,该特征在诸如应用服务器客户希望将分区从例如他们已成功开发和测试的测试环境移动到生产环境之类的场景中是有用的。

[0069] 图6示出了根据实施例的、在应用服务器、云或其它环境中可移植分区的使用。如图6中所示,根据实施例,第一域或源域400(例如,测试域)可以包括一个或多个分区,该一个或多个分区包括与分区配置404以及一个或多个分区特定的资源组、应用或资源406相关联的特定分区A 402。

[0070] 可选地,分区可以引用资源组模板410。每个资源组或包括该资源组的分区可以与定义该资源组或分区应当被定向和部署到的目标(诸如源域中的虚拟目标412)的虚拟目标(例如,虚拟主机)信息相关联。

[0071] 根据实施例,可能期望将分区导入到另一第二域或目标域420(例如,生产域)中,其中移植的(ported)分区A 422可以类似地与分区配置424和一个或多个分区特定的资源组、应用或资源426相关联,可选地引用资源组模板430,并且与定义该资源组或分区应当被定向和部署到的目标(诸如目标域中的虚拟目标432)的虚拟目标(例如,虚拟主机)信息相关联。

[0072] 根据实施例,在分区档案的创建或导出440期间,创建分区档案442,分区档案442包括分区配置444、资源组或资源组模板信息446以及可选地包括应用二进制文件448。根据实施例,分区档案可以是例如文件的zip类型。

[0073] 根据实施例,分区档案还可以包括允许访问与分区相关联的可导出或可配置值452的JSON接口450(例如,JSON文件)。在将分区档案部署或导入454到目标域期间,包括与分区相关联的可导出或可配置值在内的分区档案中的信息被用来配置分区以在目标域中使用。

[0074] 由于系统配置MBean可以随时间改变,因此系统可以将关于系统或分区属性或MBean的注释识别为例如注释@portable或@ExportCustomizableValue,这指示那些MBean应当被暴露为与分区相关联的可导出或可配置值。

[0075] 根据实施例,由于应用二进制文件可以是重量级的,所以不将这些二进制文件捆绑在分区档案中可能是优选的,因此代替地可以使用export-partition(导出分区)命令来在分区档案中包括或排除二进制文件。然后,目标域中的用户可以指定JSON文件中的应用二进制文件的位置,该位置可以是与源域的位置不同的位置。

[0076] 图7进一步示出了根据实施例的、在应用服务器、云或其它环境中可移植分区的使用。如图7中所示,根据实施例,并且如上文所描述的,分区档案包括允许访问与分区相关联的可导出或可配置值的JSON接口。

[0077] 在将分区档案部署或导入到目标域期间,可以从管理员或从另一个源接收经修改的可配置值460,该经修改的可配置值460修改与分区相关联的可导出或可配置值或者以其它方式和与分区相关联的可导出或可配置值一起使用,并且被用来提供经修订的分区配置462以供在目标域中使用。

[0078] 一些分区配置信息可以被限制到特定域,但是用户应当能够修改它以供在不同的域中使用。例如,非用户可修改的、某一域下的一组配置MBean可以与用户可修改的分区中的类似的一组MBean相关联。

[0079] 基于由管理员提供的任何配置改变,分区子部件可以在目标域中重新创建它们的环境。如果源域中的分区包括具有一个应用的资源组,则当该分区被导入到目标域中时,该资源组内的应用的配置和部署可以由系统自动执行,以适合目标域。

[0080] 例如,根据实施例,与源域相比,JSON文件可以暴露允许用户将不同的PDB与目标域中的分区相关联的属性。在导入期间,系统可以检查与分区档案相同的文件夹中的JSON文件,并且,如果它发现由用户进行的任何改变,则它使用那些改变来重写目标域中的配置默认值。

[0081] 图8示出了根据实施例的在应用服务器、云或其它环境中使用可移植分区的方法。如图8中所示,根据实施例,在步骤480处,提供应用服务器环境,包括用于执行软件应用的域以及一个或多个分区和资源组,其中每个分区提供域的管理和运行时细分并且可以包括一个或多个资源组,并且其中每个资源组是可以引用资源组模板的可部署应用或资源的集合。

[0082] 在步骤482处,接收从源域(例如,测试域)导出特定分区以用于导入到目标域(例如,生产域)中的请求。

[0083] 在步骤484处,创建分区档案,包括可以被用于为分区中的每个资源组和资源生成分区特定的配置元素的分区配置和其它信息(包括来自任何被引用的资源组模板的分区配置和其他信息),其中分区档案包括允许修改其中的可导出可配置值的接口。

[0084] 在步骤486处,系统可以结合分区档案使用针对可导出可配置值的任何接收到的修改以将特定分区部署到目标域,包括应用任何经修订的分区配置以供在该目标域中使用。

[0085] 示例实施方式

[0086] 为了说明的目的,下面提供描述示例性实施例的附加细节。

[0087] 导出分区

[0088] 如上文所描述的,当从源域导出分区时,它产生分区档案。根据实施例,以下可以被包括在分区档案中:分区配置;分区中包含的任何资源组;由那些资源组参考的任何资源组模板,以及分区文件系统的内容。如上文所描述的,可选地,用于部署到分区的应用的应用二进制文件和配置也可以包括在分区档案中。在一些实例中,不可能消除对域的所有依赖性。分区可以引用的域中的配置元素的一些示例包括目标和安全领域。

[0089] 导入分区

[0090] 在一些实例中,当导入分区档案时,系统管理员可能需要更新对域(诸如目标和安全领域)的依赖性,并且还可选地更新分区配置中的其它属性以使其有效。可能需要更新的一些项包括分区特性、JDBCSystemResourceOverrideMBean(JDBC系统资源重写MBean),以及ResourceGroup(资源组)中的其它资源。

[0091] 在导入分区时资源组模板发生冲突

[0092] 在一些实例中,分区档案可以包含与源域中的其它分区共享的资源组模板。在将分区档案导入目标域中时,有可能共享的资源组模板可能已经存在于目标域中-可能来自较早的分区导入。根据实施例,导入操作将通过执行两个树(现有资源组模板和新资源组模板)的深度树比较来检测现有资源组模板是否与档案中包含的资源组模板相同。如果它们相同,则将使用现有资源组模板。否则,如果它们不相同,则导入将失败并且要求用户重试导入,指示将被用来创建新资源组模板的资源组模板的新名称,而不是使用现有资源组模板。

[0093] 配置数据的类别

[0094] 根据实施例,存在不同类别的分区配置,这些不同类别的分区配置中的每个需要稍微不同的处理:

[0095] 在导入时不会改变:这些是从源域逐字复制到目标域的属性。如果它们需要改变,则管理员可以稍后在导入分区之后稍后进行改变。示例包括WorkManager设置。

[0096] 在导入时可能改变:这些是在导入时很可能改变、并且如果它们没有为了新域而被正确地设置则将生成验证错误的属性。这些常常是参考域中(分区外部)的配置元素的属性。示例包括对Targets和SecurityRealms的引用。

[0097] 在导入时可能会改变或者可能不会改变:这些是我们只是不知道的属性,但是如果它们确实改变,则它们应当可能在导入期间(而不是在导入之后)被改变,因为如果它们是错误的,则它们可以生成验证错误或不良行为。示例包括

JDBCSystemResourceOverrideMBean属性(URL、用户、passwd等)。

[0098] 与生命周期管理集成

[0099] 当分区被导入到新域中时,一些数量的外部系统可能需要被配置为知道新导入的分区。例如,OTD可能需要被配置为对新分区中的应用进行负载均衡。

[0100] 分区档案格式

[0101] 根据实施例,分区档案可以是具有如表1所示的内容的zip文件:

[0102]

分区档案内容	描述
partition-config.xml	这包含来自 config.xml 的分区配置和资源组模板配置
MANIFEST.MF	清单 (manifest) 文件将包括用于档案的时间戳和域信息
<PartitionName>-attributes.json	这包含在导入分区时可以由管理员潜在地改变的 MBean 属性/特性。除了被包含在档案中,这个文件的副本还在文件系统中挨着档案放置,以使它更容易在导入时更新和提供
expPartSecret	这包含用于加密和加密属性的解密的加密密钥

[0103]

domain/config/resource-group-templates /<resource_template_name>/*	如果存在与这个分区相关联的资源组模板，则与资源组模板相关的文件被复制到这里
domain/config/partitions/<partition-name>/*	包括在 domain/config/partitions/<partition-name>/config/* 下的所有配置文件
pfs/config/*	这包含<partition file system>/config dir 内容。这还将包括系统资源（JMS、JDBC 等）描述符文件
domain/upload/resource-group-templates /<resource_template_name>/<application_name>/	资源组模板级别二进制文件
pfs/upload/<application_name>/.	资源组级别应用二进制文件

[0104] 表1

[0105] exportPartition

[0106] 根据实施例，暴露exportPartition的公共层是PortablePartitionManagerMBean。

[0107] public ImportExportPartitionTaskMBean exportPartition(String partitionName,String expArchPath,Boolean includeAppsNLibs)throws Exception;

[0108] public ImportExportPartitionTaskMBean exportPartition(String partitionName,String expArchPath,Boolean includeAppsNLibs,String keyFile)throws Exception;

[0109] public ImportExportPartitionTaskMBean exportPartition(String partitionName,String expArchPath)throws Exception;

[0110] public ImportExportPartitionTaskMBean exportPartition(String partitionName,String expArchPath,String keyFile)throws Exception;其中：

[0111] partitionName是要导出的分区的名称。

[0112] expArchPath是如由管理服务器(而不是客户端)看到的到要创建的档案文件的完整路径。除了创建档案，exportPartition还将<PartitionName>-attributes.json文件的副本放在与分区档案相同的目录中。

[0113] includeAppsNLibs是影响应用二进制文件如何被处理的布尔值。默认值为真(true)。

[0114] keyFile是包含要用作用于加密属性的加密密钥的字符串的用户提供的文件。

[0115] @ExportCustomizableValue注释

[0116] 根据实施例,默认地,分区中的所有配置属性被逐字地保存在分区档案中,并且按原样(即,根据上文所描述的“不改变”类别)被导入。但是,如上文所描述的,在一些情况下,一些属性可能需要在导入期间被改变。根据实施例,为了指示这一点,可以使用@ExportCustomizableValue注释。如果配置属性具有@ExportCustomizableValue注释,那么除了将该属性保存到分区档案config.xml之外,该属性还被保存在分区档案中的<PartitionName>-attributes.json文件中。这指示用于该属性的值应当在导入时被提示。

例如:

[0117] @ExportCustomizableValue

[0118] public interface JDBCSystemResourceOverrideMBean extends ConfigurationMBean{...}

[0119] 在上面的示例中,JDBCSystemResourceOverrideMBean的所有属性连同其当前值(saveDefault默认值为true)都被复制到<PartitionName>-attributes.json。

[0120] public interface PartitionMBean extends ConfigurationMBean{

[0121] ...

[0122] @ExportCustomizableValue(saveDefault=false)

[0123] RealmMBean getRealm();

[0124] ...

[0125] }

[0126] 在上面的示例中,分区的realm(领域)属性将被保存到<PartitionName>-attributes.json而没有当前值。示例<PartitionName>-attributes.json文件可以是:

[0127] {

[0128] "partition":{

[0129] "name":"P1",

[0130] "jdbc-system-resource-override":{

[0131] "URL":"url.com",

[0132] "Id":"0",

[0133] "name":"test123",

[0134] "CachingDisabled":"false",

[0135] "Registered":"false",

[0136] "User":"test123",

[0137] "DynamicallyCreated":"false",

[0138] "DataSourceName":"test-source"

[0139] },

[0140] "partition-properties":[

[0141] {

```
[0142]   "name": "Property1",
[0143]   "value": "Value1"
[0144] },
[0145] {
[0146]   "name": "Property2",
[0147]   "value": "Value2"
[0148] }
[0149] ]
[0150] },
[0151] "resource-group-template": {
[0152]   "name": "RGT1",
[0153]   "jdbc-system-resource": {
[0154]     "name": "jdbc1",
[0155]     "descriptor-file-name": "jdbc/P1DB1-8882-jdbc.xml"
[0156]   }
[0157]   "jms-server-resource": {
[0158]     "name": "jms1",
[0159]     "PagingDirectory": ""
[0160]   }
[0161] }
[0162] }
```

[0163] importPartition

[0164] 根据实施例,暴露importPartition的公共层是
PortablePartitionManagerMBean。

```
[0165] public ImportExportPartitionTaskMBean importPartition(String
archiveFileName,boolean createNew)throws Exception;
```

```
[0166] public ImportExportPartitionTaskMBean importPartition(String
archiveFileName,boolean createNew,String keyFile)throws Exception;
```

```
[0167] public ImportExportPartitionTaskMBean importPartition(String
archiveFileName,String partitionName,Boolean createNew)throws Exception;
```

```
[0168] public ImportExportPartitionTaskMBean importPartition(String
archiveFileName,String partitionName,Boolean createNew,String keyFile)throws
Exception;
```

[0169] 其中:

[0170] archiveFileName:如由管理服务器(而不是客户端)所看到的到要导入的档案的完整路径.importPartition还将在与分区档案相同的目录中查找文件<PartitionName>-attributes.json.如果它被找到,则该文件中的值将重写分区档案中的值。

[0171] createNew:(见下文)。

[0172] partitionName:新创建的分区名称.默认为分区档案中指定的名称。

[0173] keyFile:是包含在导出期间用作用于加密属性的加密密钥的字符串的用户提供的文件。

[0174] 连同在config.xml中重新创建分区配置,需要为新分区部署应用和系统资源。

[0175] 管理员可以向<PartitionName>-attributes.json文件提供适合于目标域的经修改的属性。

[0176] createNew Flag (importPartition)

[0177] 根据实施例,在执行importPartition时,这个标记具有以下语义:

[0178] 没有指定标记:如果存在具有相同名称(或者,如果支持深度比较,则具有不是“相同RGT”的相同名称)的现有RGT,则失败。在这种情况下,导入失败,并显示类似这样的消息:“已经存在具有名称CompanyRGT的RGT。如果想要使用域中已经存在的CompanyRGT,则请用createNew=false来重新运行此命令。如果想要忽略现有CompanyRGT并且使用分区档案和attributes.json文件中的定义来创建新的CompanyRGT,则请用createNew=true来重新运行此命令”。

[0179] false:如果在导入期间存在冲突,则使用现有RGT。

[0180] true:如果存在冲突,则创建新的RGT。

[0181] 目标域中的任何现有RGT从未被修改。

[0182] 应用/库二进制文件的导出

[0183] 根据实施例,当分区被导出时,应用二进制文件是否包括在分区档案中由关于导出操作的“includeApps”选项的值来确定。如果布尔值“includeApps”为真,则二进制文件被包括,否则它们被排除。

[0184] 不管“includeApps”设置为真或假,<partition>-attributes.json都将具有用于范围限定到分区和被导出的分区内的资源组的每个应用的条目。应用属性“source-path”,“plan-path”、“staging-mode”对终端用户最为重要。

[0185] 应用/库二进制文件的导入

[0186] 根据实施例,在importPartition期间应用如何被部署取决于若干因素,例如应用二进制文件是否存在于<partition-name>.zip中;或<partition-name>-attributes.json中的分级模式(staging-mode)属性。

[0187] 如果应用二进制文件存在于<partition-name>.zip中,则在importPartition期间,(来自zip的)这个应用二进制文件被使用。这个二进制文件被复制到该应用的父亲(假设是ResourceGroupTemplate或ResourceGroup)的上传目录。无论导出分区时的分级模式是什么,应用的分级模式都设置为“STAGE”。

[0188] 如果应用二进制文件不存在于<partition-name>.zip中,则在importPartition期间,从<partition-name>-attribute.json文件读取“staging-mode”属性,然后采取以下操作:

[0189] 如果分级模式是“STAGE”,则抛出暗示如果分段模式是阶段(stage),则应用二进制文件应当已经在导出期间包括在zip文件中的错误消息。

[0190] 如果分级模式是“NOSTAGE”或“EXTERNAL”,则来自<partition-name>-attributes.json的“source-path”属性被读取,并且用作为分段模式的“NOSTAGE”或“EXTERNAL”(从<partition-name>-attributes.json文件读取的任何内容)和读取的源路

径来部署应用。如果应用二进制文件在读取的源路径处不存在,则抛出错误并且停止 importPartition。

[0191] 加密属性的导出

[0192] 根据实施例,域(Domain)树中的每个加密属性可以使用(保存在 SerializedSystemInd.dat中的)特定于域的密钥来加密,这意味着加密属性不能被原样导出,因为密钥 SerializedSystemIni.dat 对于每个域是不同的。因此,以下步骤被用来处理加密数据的导出:

[0193] 在导出期间生成新的隐秘密钥(Secret key)。

[0194] 使用隐秘密钥加密 partition-config.xml 中的所有加密属性以及作为 <Partition-name>.zip 的部分的任何 SystemResource 描述符。

[0195] 使用盐(salt)和第二个密钥来加密隐秘密钥并将其保存到作为导出的 zip <Partition-name>.zip 的一部分的文件 expPartSecret 中。

[0196] 加密属性的导入

[0197] 在读取整个分区并且根据 <Partition-name>-attributes.json 改变属性之后的导入期间,使用以下步骤来处理加密属性:

[0198] 读取作为导出的 zip 的一部分的隐秘密钥 expPartSecret。

[0199] 使用第二密钥解密隐秘密钥,该第二密钥或者是 WLS 源中的默认密钥,或者是由用户使用 keyFile 选项提供的用于导入的密钥。用户提供的密钥必须与导入时使用的密钥相匹配。

[0200] 使用从 expPartSecret 读取的密钥解密(上文读取和经修改的)分区 MBean 中的所有加密属性。

[0201] 利用用于导入域的域特定的密钥(SerializedSystemIni.dat)加密分区 MBean 中的所有加密属性。

[0202] 可以使用包括根据本公开的教导编程的一个或多个处理器、存储器和/或计算机可读存储媒介的一个或多个常规的通用或专用数字计算机、计算设备、机器或微处理器来方便地实现本发明。如对软件领域的技术人员将明显的,基于本公开的教导,熟练的程序员可以容易地准备适当的软件编码。

[0203] 在一些实施例中,本发明包括计算机程序产品,该计算机程序产品是具有存储在其上/其中的可以被用于对计算机进行编程以执行本发明的过程中的任何过程的指令的非暂态存储介质或计算机可读介质(媒介)。存储介质可以包括但不限于任何类型的盘,包括软盘、光盘、DVD、CD-ROM、微型驱动器以及磁光盘、ROM、RAM、EPROM、EEPROM、DRAM、VRAM、闪存存储设备、磁卡或光卡、纳米系统(包括分子存储器 IC),或适合于存储指令和/或数据的任何类型的媒介或设备。

[0204] 出于说明和描述的目的提供了本发明的上述描述。它并不旨在是详尽的或者将本发明限制到所公开的精确形式。许多修改和变化对于本领域技术人员来说将是明显的。实施例被选择和描述以便于最佳地解释本发明的原理及其实际应用,从而使得本领域其他技术人员能够对于各种实施例以及利用适合所设想的特定使用的各种修改来理解本发明。旨在由以下权利要求和它们的等价物来限定本发明的范围。

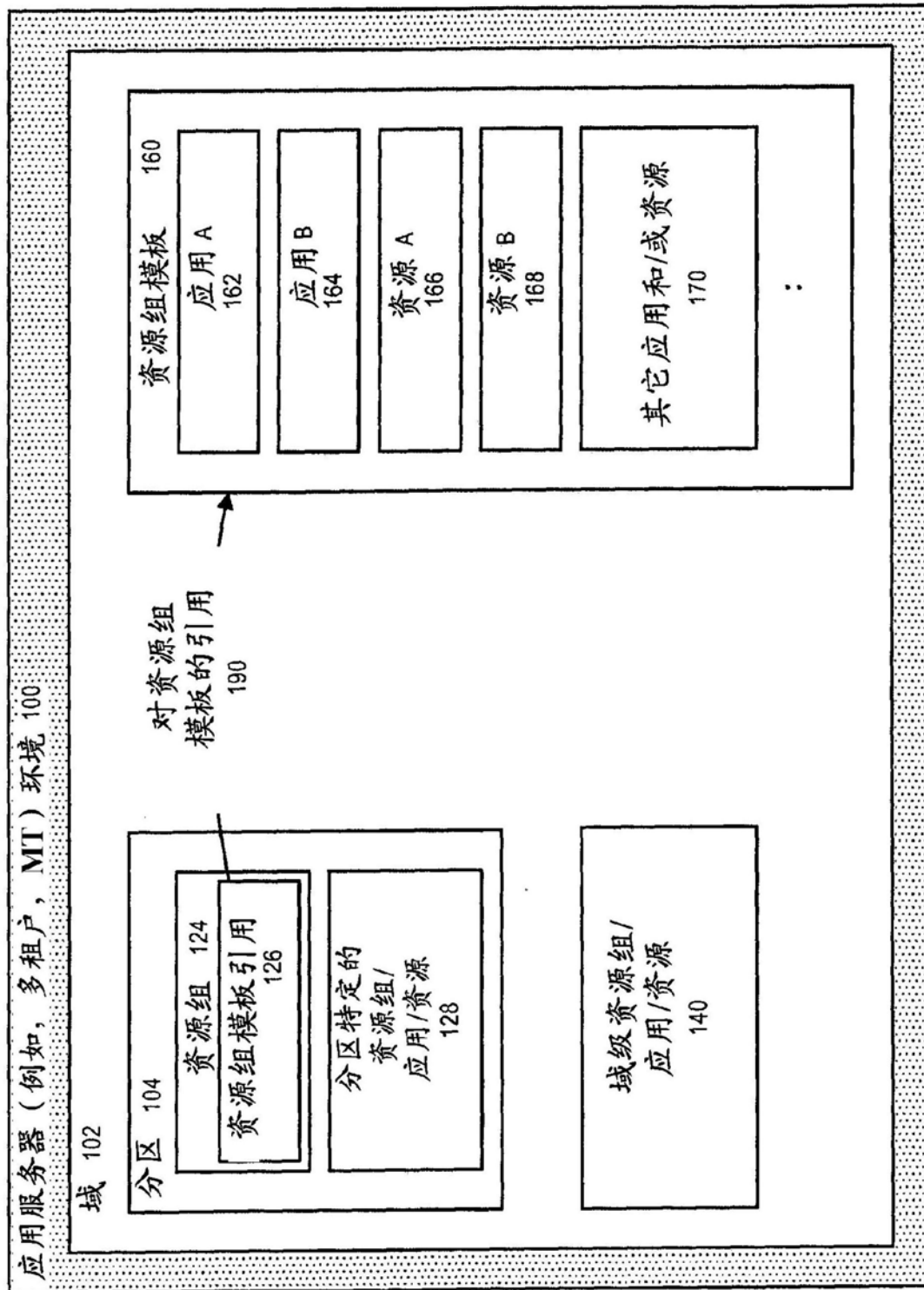


图1

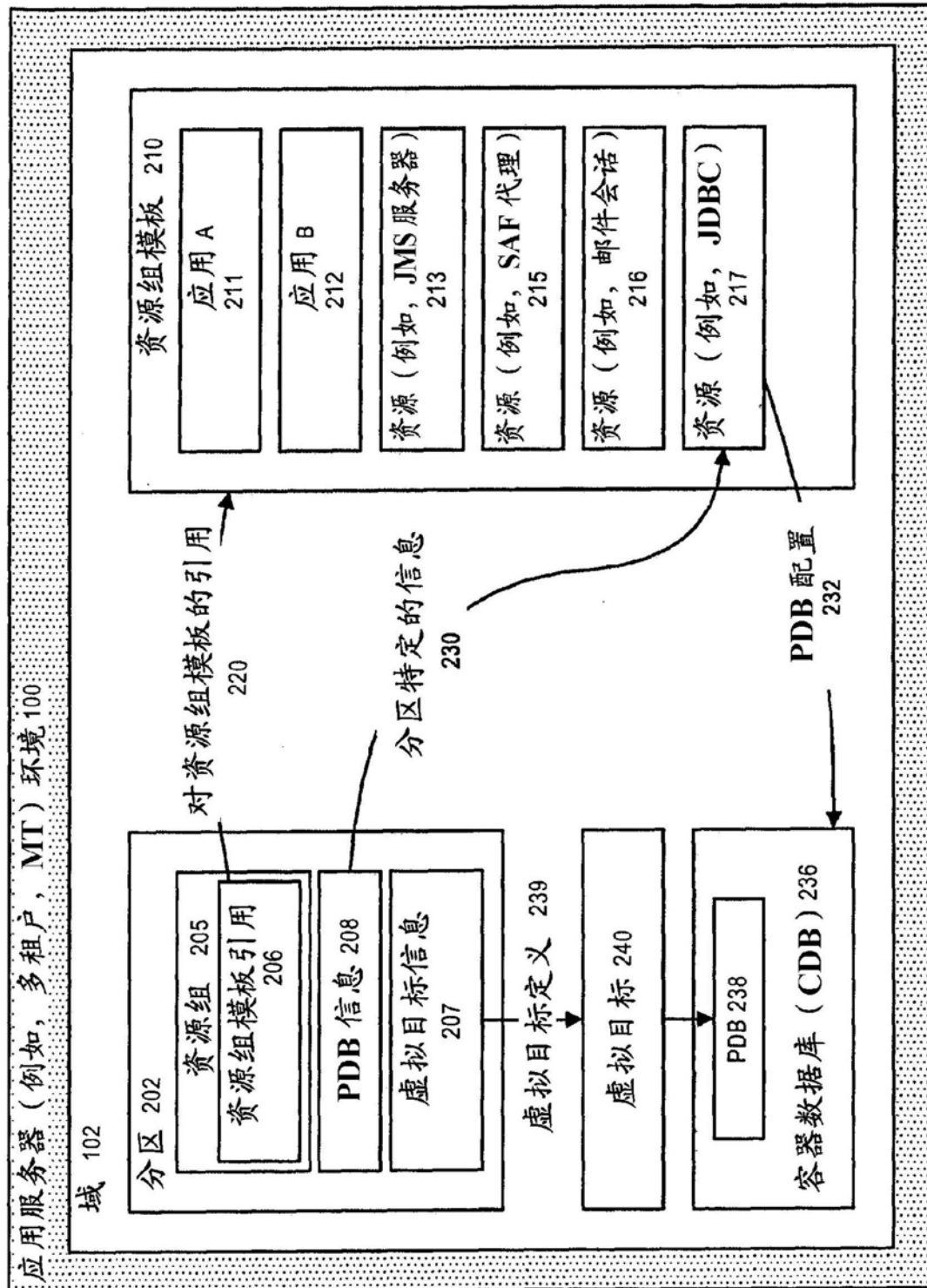


图2

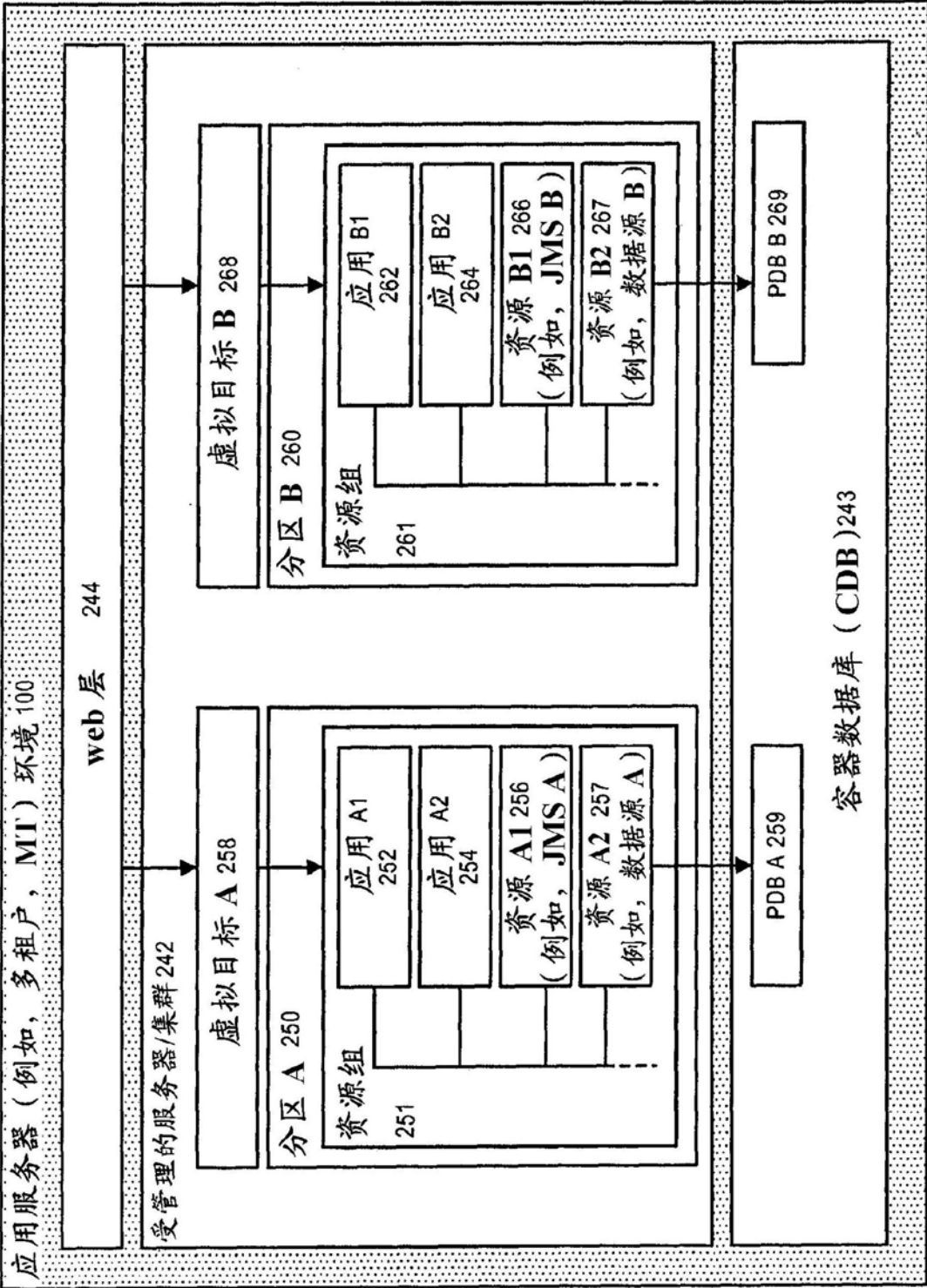


图3

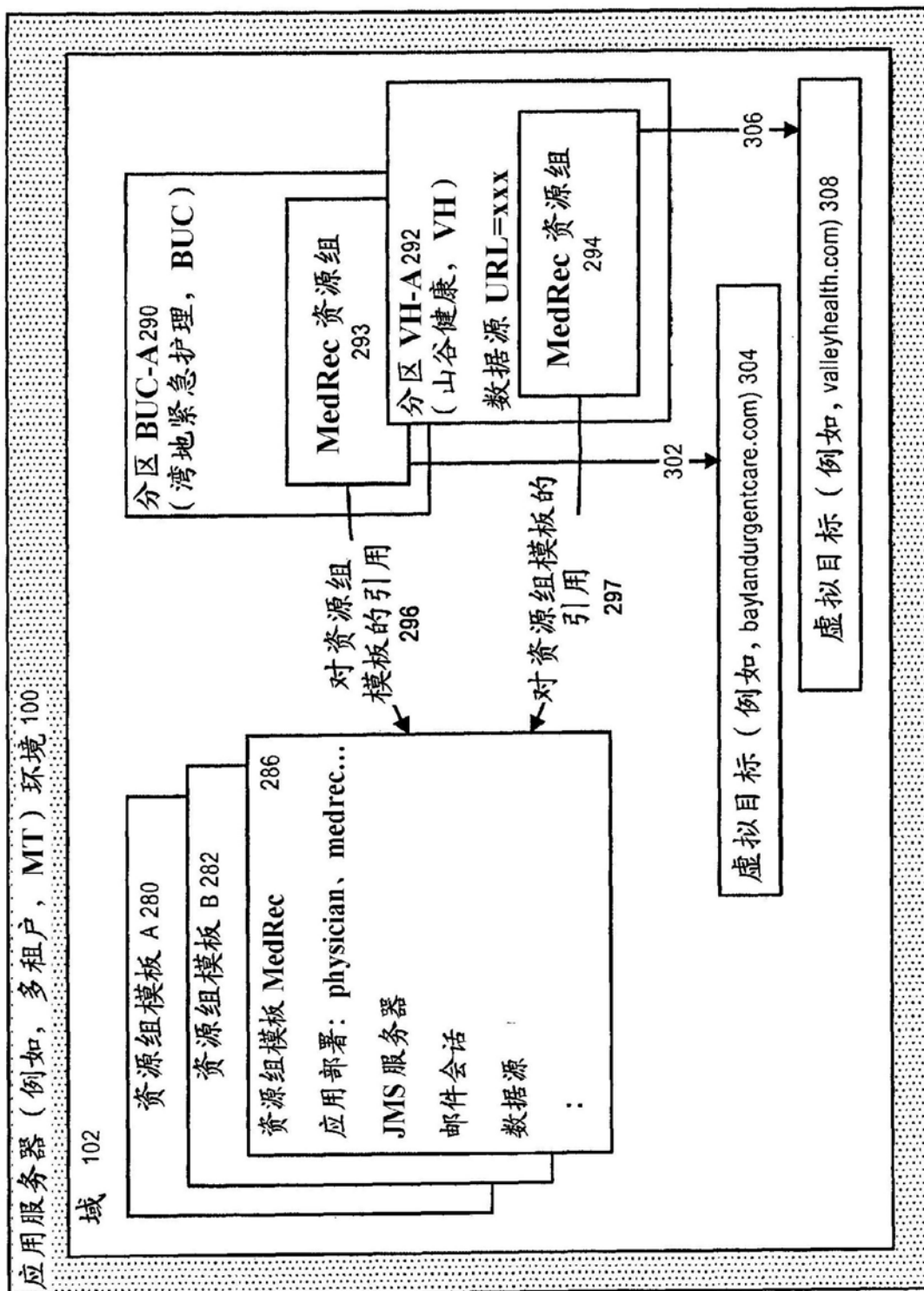


图4

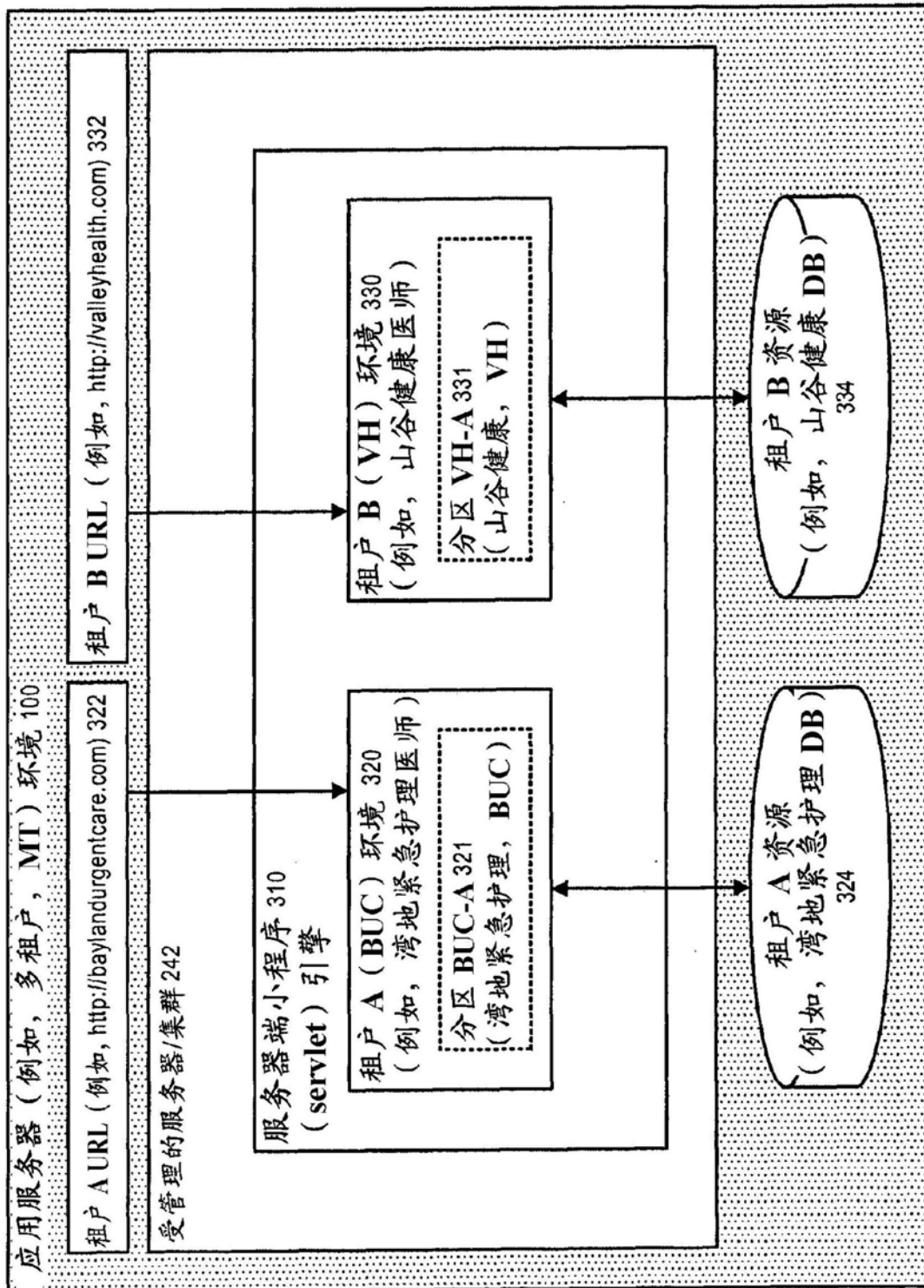


图5

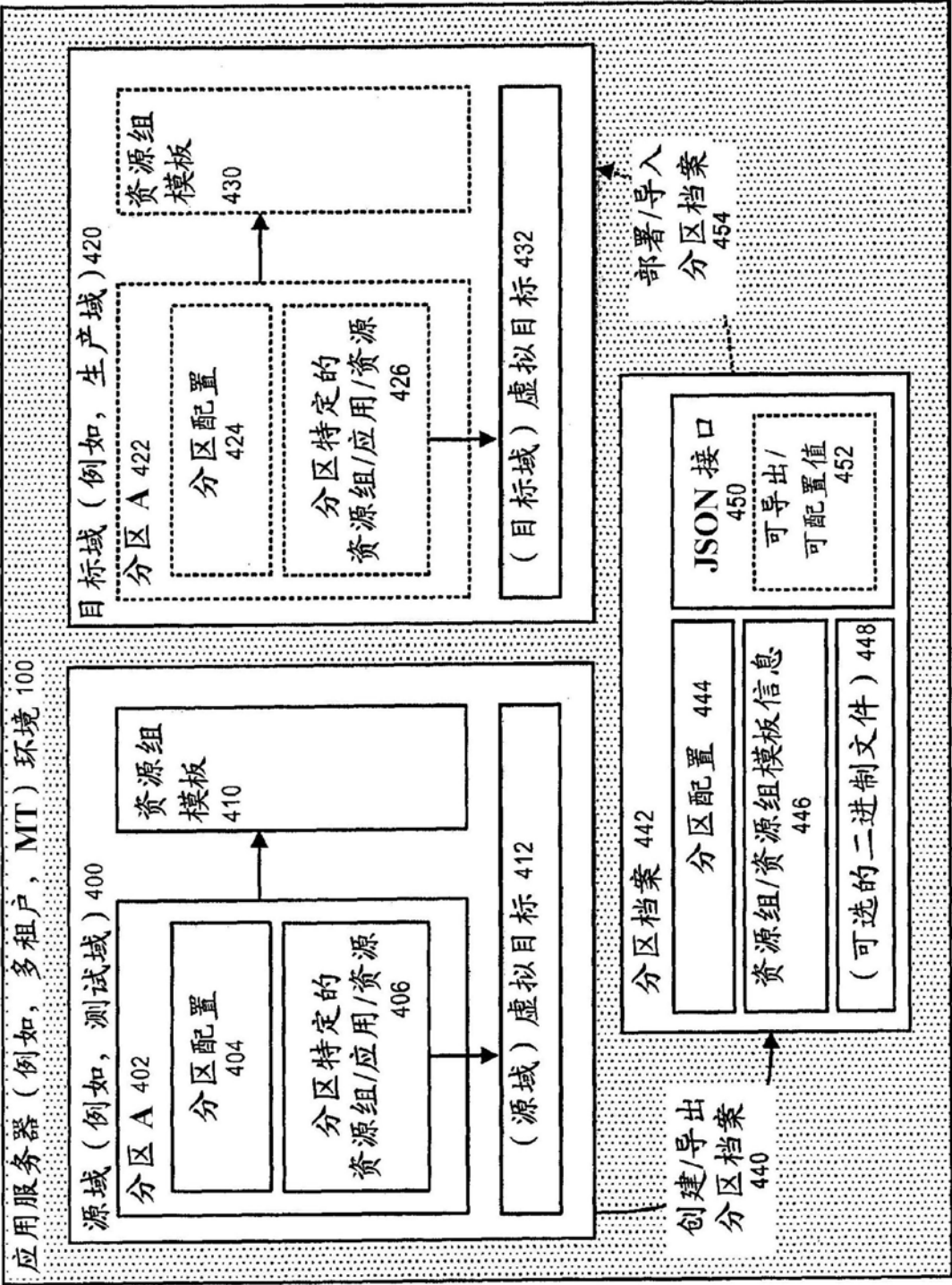


图6

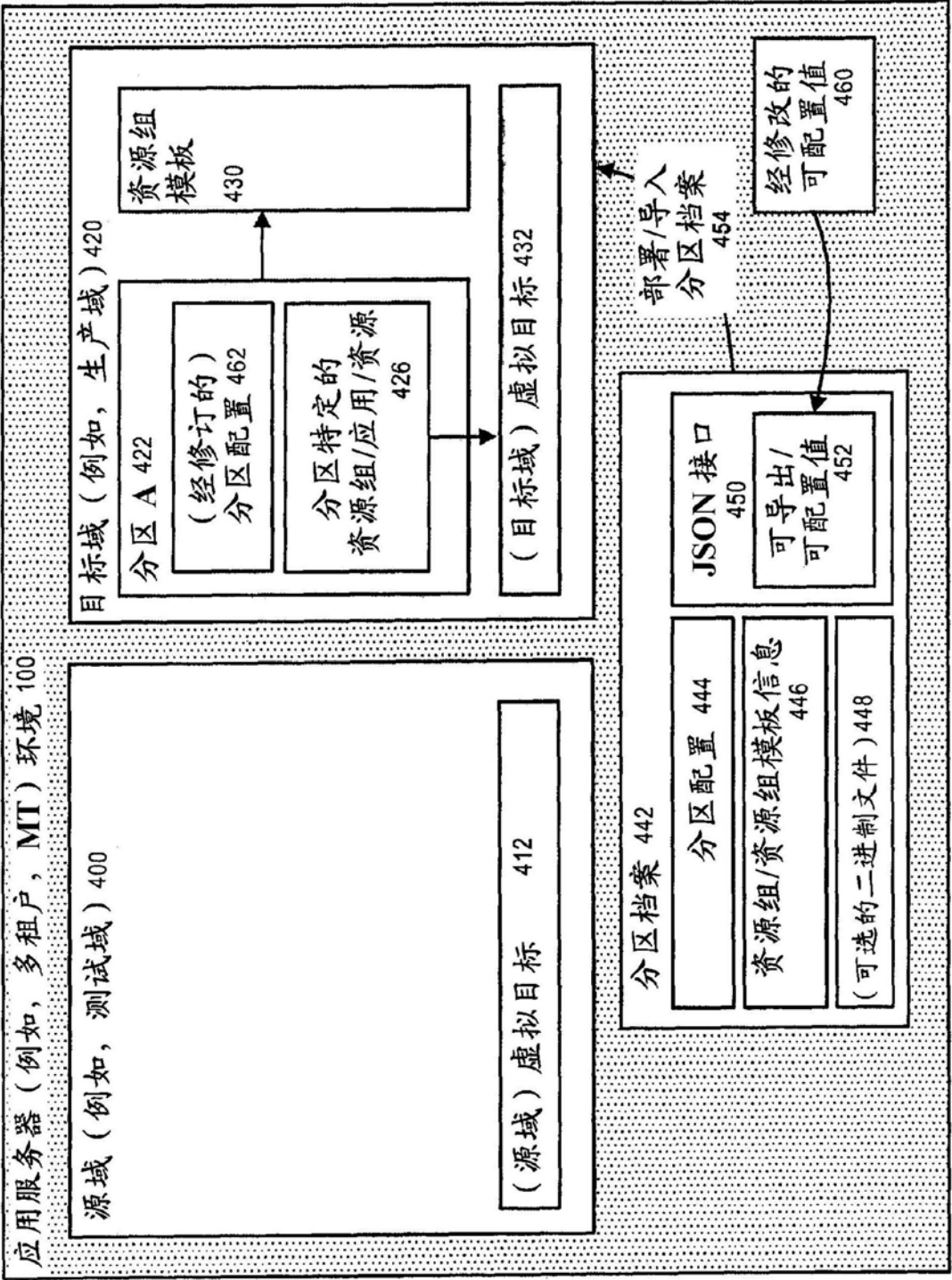


图7

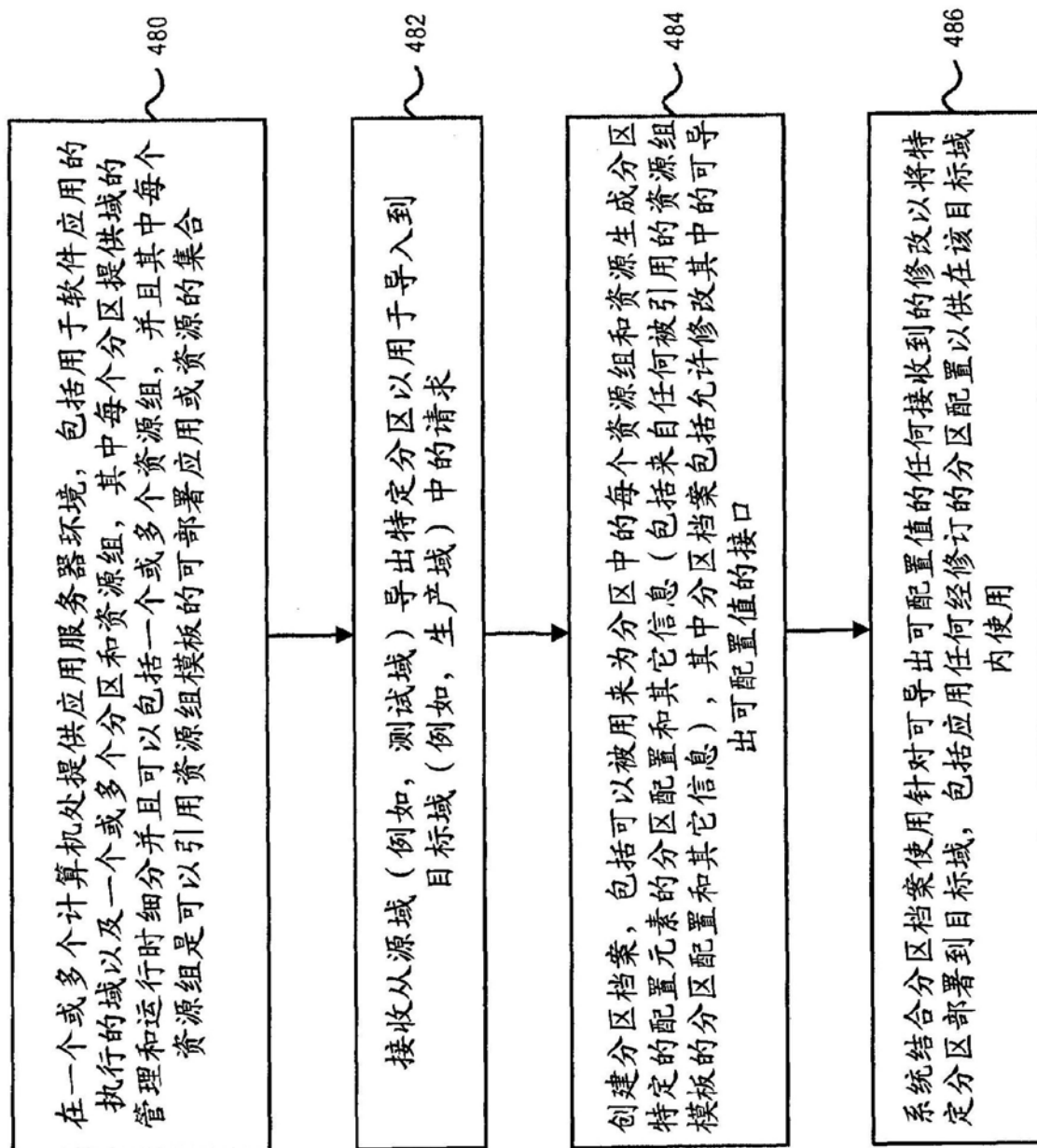


图8