



[12] 发明专利说明书

专利号 ZL 200480023352.3

[45] 授权公告日 2008年7月30日

[11] 授权公告号 CN 100407153C

[22] 申请日 2004.8.13

[21] 申请号 200480023352.3

[30] 优先权

- [32] 2003. 8. 14 [33] US [31] 60/495,368
- [32] 2003. 9. 3 [33] US [31] 60/500,096
- [32] 2003. 9. 3 [33] US [31] 60/500,050
- [32] 2003.11.21 [33] US [31] 10/718,747
- [32] 2004. 8. 12 [33] US [31] 10/917,687
- [32] 2004. 8. 12 [33] US [31] 10/917,873
- [32] 2004. 8. 12 [33] US [31] 10/917,953
- [32] 2004. 8. 12 [33] US [31] 10/917,661
- [32] 2004. 8. 12 [33] US [31] 10/918,055
- [32] 2004. 8. 12 [33] US [31] 10/918,056
- [32] 2004. 8. 12 [33] US [31] 10/918,054
- [32] 2004. 8. 12 [33] US [31] 10/917,715

[86] 国际申请 PCT/US2004/026405 2004.8.13

[87] 国际公布 WO2005/017745 英 2005.2.24

[85] 进入国家阶段日期 2006.2.14

[73] 专利权人 甲骨文国际公司

地址 美国加利福尼亚州

[72] 发明人 拉克希米纳拉亚南·奇丹巴拉恩

詹姆斯·W·斯塔莫斯

拉金德拉·平特 埃德温娜·卢

[56] 参考文献

WO0207037A1 2002.1.24

审查员 孟宪超

[74] 专利代理机构 北京康信知识产权代理有限责
任公司

代理人 余刚

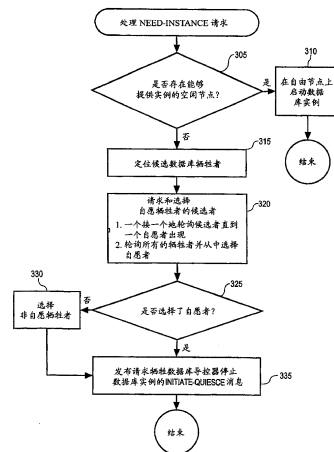
权利要求书4页 说明书29页 附图7页

[54] 发明名称

需要时节点和服务实例分配和解除分配

[57] 摘要

一种方法有效并动态地分配和解除分配数据库之间的数据库实例和节点。该方法以解决多数据库的用户和由多数据库容留的服务的方式执行该功能。



1. 一种用于管理多节点系统中的资源的方法,所述方法包括以下步骤:

所述多节点系统检测在多节点服务器上的服务级别协定的违反,其中所述多节点服务器包括运行第一节点组的服务器实例组;

响应于检测所述违反,所述多节点系统确定是否通过使用不是所述第一节点组的成员的节点容留所述多节点服务器的另一服务器实例来解决所述违反; 以及

响应于确定通过使用不是所述第一节点组的成员的节点来解决所述违反,通过执行以下步骤来尝试解决所述违反,所述步骤包括:

所述多节点系统寻找另一节点以用来容留所述多节点服务器的第二服务器实例,所述另一节点在被找到时容留第三服务器实例;

所述多节点系统将在所述第三服务器实例上为一个或多个客户机建立的至少一个会话迁移至容留在所述多节点系统上的另一服务器实例; 以及

所述多节点系统启动在所述另一节点上的所述多节点服务器的所述第二服务器实例。

2. 根据权利要求1所述的方法, 其中

所述尝试解决所述违反的步骤进一步包括所述多节点系统在启动所述服务器实例之前关闭所述第三服务器实例。

3. 根据权利要求1所述的方法,其中所述寻找另一节点的步骤进一步包括:

产生候选者组,该候选者组对应于由不同于所述第一节点组的另一节点组容留的另一多节点服务器;以及

从所述从其解除分配节点的候选者组中选择牺牲者。

4. 根据权利要求3所述的方法,其中

所述步骤进一步包括发送请求至对应于自愿停止节点上的服务器实例的所述候选者组的候选者的多节点服务器;以及

选择牺牲者包括从多个多节点服务器的候选者选择牺牲者,所述多节点服务器响应于所述的停止请求发送消息以自愿停止。

5. 根据权利要求4所述的方法,其中所述发送请求至多节点服务器的步骤包括将所述请求广播给所述候选者组。

6. 根据权利要求4所述的方法,其中所述从多个多节点服务器的候选者选择牺牲者的步骤包括基于一个或多个因素选择牺牲者,所述因素至少包括以下之一:

优先级数据,由用户提供以及表示所述多个多节点服务器的一个候选者相对于所述多个多节点服务器的其它候选者的优先级;以及

停止成本,与停止节点上的服务器实例相关联。

7. 根据权利要求2所述的方法,其中所述尝试解决所述违反的步骤包括:

将请求发送至第二多节点服务器以准备服务器实例停止;以及

从所述第二多节点服务器接收信息，指示服务器实例准备被停止。

8. 根据权利要求7所述的方法，其中

所述指示服务器实例准备的消息进一步包括识别所述第二服务器实例和节点的数据。

9. 一种用于管理多节点系统中的资源的方法，所述方法包括以下步骤：

多节点服务器，运行在所述多节点系统的多个节点上，所述多节点服务器包括运行在所述多个节点上的多个服务器实例；

所述多节点服务器接收确认是否至少一个服务器实例可以被释放的请求；

响应于接收所述请求，基于性能的指标，选择在其上停止为所述多节点服务器的客户机执行工作的所述多个服务器实例的特定服务器实例；

使所述特定服务器实例停止为所述客户机执行工作；以及

其中使所述特定服务器实例停止为所述客户机执行工作包括将在所述特定服务器实例上为至少一个所述客户机建立的至少一个会话迁移至容留在所述多节点服务器上的另一服务器实例。

10. 根据权利要求9所述的方法，其中所述性能的指标是基于在其上容留服务器实例的节点的CPU使用率。

11. 根据权利要求9所述的方法，其中

所述特定服务器实例具有为所述多节点服务器的客户机建立的会话；

所述步骤进一步包括确定所述会话是否可以从所述特定服务器实例迁移到所述多个服务器实例的另一个；以及

所述使所述特定服务器实例停止为所述客户机执行工作的步骤只有确定在所述会话可以从所述特定服务器实例迁移时被执行。

需要时节点和服务实例分配和解除分配

相关申请

本申请要求于 2003 年 8 月 14 日提交的美国临时申请第 60/495,368 号题为“Computer Resource Provisioning”的优先权，其内容结合于此作为参考；本申请要求于 2003 年 9 月 3 日提交的美国临时申请第 60/500,096 号题为“Service Based Workload Management and Measurement in a Distributed System”的优先权，其内容结合于此作为参考；本申请要求于 2003 年 9 月 3 日提交的美国临时申请第 60/500,050 号题为“Automatic And Dynamic Provisioning Of Databases”的优先权，其内容结合于此作为参考。

本申请涉及以下美国申请：

2004 年 8 月 12 日 Lakshinarayan Chidambaran 等人提交的美国申请第 10/917,687 号题为“On Demand Node and Server Instance Allocation and De-Allocation”（代理人卷号第 50277-2382 号），其内容结合于此作为参考。

2003 年 11 月 21 日提交的美国申请第 10/718,747 号题为“Automatic and Dynamic Provisioning of Databases”（代理人卷号第 50277-2343 号），其内容结合于此作为参考。

2004 年 8 月 12 日 Benny Souder 等人提交的美国申请第 10/917,873 号题为“Hierarchical Management of the Dynamic

Allocation of Resources in a Multi-Node System” (代理人卷号第 50277-2382 号), 其内容结合于此作为参考。

2004 年 8 月 12 日 Sanjay Kaluskar 等人提交的美国申请第 10/917,953 号题为 “Transparent Session Migration Across Servers” (代理人卷号第 50277-2383 号), 其内容结合于此作为参考。

2004 年 8 月 12 日 Lakshminarayanan Chidambaran 等人提交的美国申请第 10/917,661 号题为 “Calculation of Service Performance Grades in a Multi-node Environment That Hosts the Services” (代理人卷号第 50277-2410 号), 其内容结合于此作为参考。

2004 年 8 月 12 日 Lakshminarayanan Chidambaran 等人提交的美国申请第 10/918,055 号题为 “Incremental Run-Time Session Balancing in a Multi-Node System” (代理人卷号第 50277-2411 号), 其内容结合于此作为参考。

2004 年 8 月 12 日 Lakshminarayanan Chidambaran 等人提交的美国申请第 10/918,056 号题为 “Service placement for Enforcing Performance and Availability Levels in a Multi-Node System” (代理人卷号第 50277-2412 号), 其内容结合于此作为参考。

2004 年 8 月 12 日 Lakshminarayanan Chidambaran 等人提交的美国申请第 10/918,054 号题为 “Recoverable Asynchronous Message Driven Processing in a Multi-Node System” (代理人卷号第 50277-2414 号), 其内容结合于此作为参考。

2004 年 8 月 12 日 Carol Colrain 等人提交的美国申请第 10/917,715 号题为 “Managing Workload by Service” (代理人卷号第 50277-2337 号), 其内容结合于此作为参考。

本发明涉及以下国际申请:

2004年8月9日 Oracle International Corporation 在美国受理局提交的第 PCT/US04/25805 号题为“Automatic and Dynamic Provisioning of Databases”的国际申请(代理人卷号第 50277-2571 号), 其内容结合于此作为参考。

2004年8月13日 Oracle International Corporation 在美国受理局提交的第 PCT/US04/26389 号题为“Hierarchical Management of the Dynamic Allocation of Resources in a Multi-Node System”的国际申请(代理人卷号第 50277-2592 号), 其内容结合于此作为参考。

2004年8月13日 Oracle International Corporation 在美国受理局提交的第 PCT/US04/26445 号题为“Transparent Session Migration Across Servers”的国际申请(代理人卷号第 50277-2593 号), 其内容结合于此作为参考。

2004年8月13日 Oracle International Corporation 在美国受理局提交的第 PCT/US04/26570 号题为“Transparent Migration of Stateless Sessions Across Servers”的国际申请(代理人卷号第 50277-2594 号), 其内容结合于此作为参考。

技术领域

本发明涉及工作负荷管理, 具体而言, 涉及多节点计算机系统内的工作负荷管理。

背景技术

各企业正在寻找办法使他们的数据处理系统降低成本并提高效率。典型的企业数据处理系统是为每个企业应用程序都分配单独

的资源。预先为每个应用程序分配足够的资源以处理该应用程序的估计最大负荷。各应用程序具有不同的负荷特性；一些应用程序在白天繁忙；而另一些应用程序则在晚上繁忙；一些报告每周运行一次，而另一些报告则每月一次。因此，有大量资源能力被闲置。网格计算使利用或消除这种闲置能力成为可能。实际上，网格计算被平衡以显著地改变了计算的经济效果。

网格是多个提供处理和一定程度的共享存储的计算元件的集合；网格资源被动态分配，以满足其客户机（client）的计算需求和优先级。网格计算可显著地降低计算成本，扩大计算资源的可用性，以及实现更高的生产率和更高的质量。网格计算的基本思想是将计算看作一种公用事业（utility），类似于电网或电话网。网格的客户机不关心其数据在何处或者在何处执行计算。客户机所想要的只是使计算被执行，以及当客户机需要时，将信息传递给客户机。

这类似于电力公用事业运行的方式：客户不知道发电机在何处，或者电网是如何连接的。客户只是要电力并得到了电力。目的是将计算变成一种公用事业—无处不在的商品。因此将其命名为网格。

当然，将网格计算看作一种公用事业是客户机端的看法。从服务器侧或者现场后面来看，网格是关于资源分配、信息共享、和高可用性。资源分配确保所有那些需求或请求资源的都如愿以偿。资源不会被持续闲置，而请求却得不到服务。信息共享确保在信息客户机和应用程序被需要的地方和时刻，所需的信息客户机和应用程序可用。高可用性确保了所有数据和计算必须一直运转—正如公用事业公司必须一直提供电力。

用于数据库的网格计算

可以从网格计算获益的一个计算机技术领域是数据库技术。网格可以支持多数据库以及根据要求动态分配和重新分配资源，以支持每个数据库的当前需求。当对数据库的需求增长时，更多的资源分配给该数据库，而从另一数据库释放其他资源。例如，在企业网络上，数据库是由运行在网格上的一个服务器刀片上的一个数据库服务器来服务的。向数据库请求数据的用户数量增加。响应于对数据库需要的增加，从一台刀片式服务器上取消用于另一数据库的数据库服务器，向该刀片式服务器提供用于正经受增长的用户请求的数据库的数据库服务器。

用于数据库的网格计算要求以不同的级别分配和管理资源。在对应于单数据库的级别，提供给数据库用户的性能和资源的可用性以及用户之间所分配的数据库资源必须被监控，以确保每个用户的性能和资源可用性目标被满足。在数据库之间，必须管理数据库之间的网格资源的分配，以确保满足所有数据库的用户的性能和资源可用性目标。管理这些不同级别的资源的分配的工作，以及执行这种管理所需要的信息非常复杂。因此，要求提供一种机制，能够简化和有效地处理用于数据库系统的网格计算系统中的资源的管理，以及在网格内以不同级别分配资源的其它类型的系统。

一种这样的机制是在 Hierarchical Management of the Dynamic Allocation of Resources in a Multi-Node System (50277-2382)中描述的系统，该系统利用分层导控器管理不同等级上的资源。一种类型的导控器（数据库导控器）管理分配给数据库用户间的数据库的资源。例如，网格可以容留用于一个数据库的一组数据库服务器。该组中的每个数据库服务器被称作数据库实例，每个数据库实例容留用于用户和一个或多个服务的许多数据库会话。数据库导控器管理用户和服务之间的数据库可用的数据库资源的分配。

服务是被容留用于一个或多个客户机的利益的特定类型或类别的工作。作为部分服务执行的工作包括任何计算机资源的使用和花费，例如 CPU 处理时间，在易失存储器中存储和获取数据，读取和写入永久存储器（例如磁盘空间），以及网络或总线带宽的使用。例如，服务可能是为数据库服务器的客户机上的特定应用程序执行的工作。

对于数据库，数据库实例组的子集被分配以提供特定的服务。被分配以提供该服务的数据库实例在此称作容留该服务。数据库实例可能容留不止一个服务。被数据库的数据库实例容留的服务在此称作被数据库容留。

由被数据库容留的服务实现的资源的性能和可用性有时可能不满足资源的性能和可用性的要求。当出现这种情形时，另一个数据库实例和容留该数据库实例的节点可以在网格中被分配给数据库。经常，可用于分配给数据库的唯一的节点池已经用于其他数据库和服务。因而将该节点分配给数据库需要从另一个数据库解除分配节点，这样会影响由容留在另一个数据库上的服务实现的资源的性能和可用性。

基于前面所述，需要一种从节点池中解除分配已分配给一组数据库的节点的方法，为由这组数据库容留的服务的用户解决和提供性能和资源可用性。

本节中所描述的方法是可执行的方法，但不必是已被构思或执行的方法。因此，除非在此另有说明，在此部分中描述的方法不能仅因为包含在此背景技术部分中，就认为是现有技术。

发明内容

针对上述现有技术中的问题，本发明提供了一种用于管理多节点系统中的资源的方法，方法包括以下步骤：多节点系统检测在多节点服务器上的服务级别协定的违反，其中多节点服务器包括运行第一节点组的服务器实例组；响应于检测违反，多节点系统确定是否通过使用不是第一节点组的成员的节点容留多节点服务器的另一服务器实例来解决违反；以及响应于确定通过使用不是第一节点组的成员的节点来解决违反，通过执行以下步骤来尝试解决违反，步骤包括：多节点系统寻找另一节点以用来容留多节点服务器的第二服务器实例，另一节点在被找到时容留第三服务器实例；多节点系统将在第三服务器实例上为一个或多个客户机建立的至少一个会话迁移至容留在多节点系统上的另一服务器实例；以及多节点系统启动在另一节点上的多节点服务器的第二服务器实例。

根据本发明的另一方面，还提供了一种用于管理多节点系统中的资源的方法，所述方法包括以下步骤：多节点服务器，运行在多节点系统的多个节点上，多节点服务器包括运行在多个节点上的多个服务器实例；多节点服务器接收确认是否至少一个服务器实例可以被释放的请求；响应于接收请求，基于性能的指标，选择在其上停止为多节点服务器的客户机执行工作的多个服务器实例的特定服务器实例；使特定服务器实例停止为客户机执行工作；以及其中使特定服务器实例停止为客户机执行工作包括将在特定服务器实例上为至少一个客户机建立的至少一个会话迁移至容留在多节点服务器上的另一服务器实例。

根据本发明提供的方法，提供了从节点池中解除分配已分配给一组数据库的节点的方法，为由这组数据库容留的服务的用户解决和提供性能和资源可用性。

附图说明

本发明通过实例而不是限制的方式被阐明，附图中相同的附图标号表示相同的部件，其中：

图1是示出了本发明的一个实施例可在其上实现的多节点计算机系统的框图；

图2是示出了管理解决服务等级协定的违反所采用的补救的升级的过程的流程图；

图3是示出了根据本发明的实施例响应于请求以为数据库得到另一数据库实例的过程的流程图；

图4是示出了根据本发明的实施例的关闭并提出节点上的另一数据库实例的过程的流程图；

图5是示出了根据本发明的实施例的确定是否有数据库实例自愿停止的过程的流程图；

图6是示出了根据本发明的实施例用于停止数据库实例的过程的流程图；以及

图7是可用在本发明的实施例中的计算机系统的框图。。

具体实施方式

描述了管理多节点环境中的资源分配的方法和装置。在下面的描述中，出于解释的目的，许多具体的细节被阐明以提供对本发明的彻底理解。然而，显而易见，本发明可以在没有这些具体细节的情况下实施。在其它实例中，已知的结构和装置以框图的形式示出，以避免对本发明造成不必要的混淆。

这里描述的是用于有效地、动态地在数据库间分配和解除分配数据库实例和节点的方法。该方法以由多个数据库容留的解决用户和服务的方式执行这个功能。

图 1 示出了可用于实现本发明的实施例的多节点计算机系统。参照图 1，示出了集群场 (cluster farm) 101。集群场是一组节点，这些节点被组织成被称之为集群的节点组。集群在该集群中的节点之间提供了一定程度的共享存储 (例如，共享一组磁盘驱动器)。集群场 101 包括集群 110、170、和 180。每个集群容留一个或多个多节点数据库服务器，这些服务器提供并管理对数据库的访问。

集群场中的节点的形式可以是通过网络相互连接的计算机 (例如工作站、个人计算机)。可选地，节点可以是网格的节点。网格由节点构成，这些节点的形式是与机架上其它服务器刀片相互连接的服务器刀片。每个服务器刀片都是包括处理器、存储器、网络连接、以及单一主板上的有关的电子器件的计算机系统。通常，服务器刀片不包括板上存储器 (除了易失性存储器)，并且这些服务器刀片共享机架内的存储单元 (例如，共享的磁盘)，以及电源、冷却系统、和电缆。

在集群场中的集群的限定特性是通过软件控制可在场内的集群之间自动地转移该集群的节点，而无需将来自一个集群的节点有形地重新连接至另一个。集群被在此称作集群软件的应用程序控制和管理。可执行集群软件以从集群中删除节点，以及为集群提供节点。集群软件提供命令行界面，其接收来自管理人员的请求，允许管理员输入命令以为集群提供节点和从集群中删除节点。该界面还可以采取应用程序接口 (“API”) 的形式，该接口可以被在该集群场内执行的其他软件调用。集群软件使用和维持元数据，这些元数据定义了场内集群的配置，包括集群配置元数据，其定义了场中集群的拓扑结构，包括了哪些特定节点在该集群中。元数据被修

改以反映集群软件对集群场内的集群作出的改变。集群软件的实例是由 Oracle™ 开发的软件，例如 Oracle9i Real Application Clusters 或 Oracle Real Application Clusters 10g。在 Oracle9i RAC: Oracle Real Application Clusters Configuration and Internals (作者为 Mike Ault 和 Madhu Tamma, 第二版, 2003 年 8 月 2 日) 中描述了 Oracle9i Real Application Clusters。

集群和多节点数据库服务器

集群 110、170 和 180 容留一个或多个多节点数据库服务器。集群 110 容留用于数据库 150 的多节点数据库服务器，该多节点数据库服务器包括数据库实例 123、125 和 127，分别容留在节点 122、124 和 126 上。集群 110 也容留用于数据库 160 的多节点数据库服务器，该多节点数据库服务器包括数据库实例 133、135 和 137，分别容留在节点 132、134 和 136 上。

服务器，例如数据库服务器，是集成的软件组件和计算资源（例如存储器、节点和节点上的用于在处理器上执行集成的软件组件的进程）的组合，软件和计算资源的组合专用于执行代表一个或多个客户机的特定功能。在数据库管理的其它功能中，数据库服务器管理和帮助访问特定数据库，处理客户机访问数据库的请求。

来自多节点计算系统中的多个节点的资源可以被分配，以运行服务器的软件。每个软件与来自节点的资源分配的结合是在此被称作“服务器实例”或“实例”的服务器。因此，多节点数据库服务器包括可在多个节点上运行的多个服务器实例。实际上，多节点数据库服务器的多个实例可以在相同的节点上运行。多节点数据库服务器包括多个“数据库实例”，每个数据库实例在节点上运行，并控制和帮助对特定数据库的访问。数据库实例 123、125 和 127 是相同多节点数据库服务器的实例。

服务

如上所述，服务是为了一个或多个客户机的利益而容留的特定类型或种类的工作。服务的一种类型是数据库服务。集群提供用于访问数据库 **150** 的数据库服务以及用于访问数据库 **160** 的数据库服务。通常，数据库服务是由数据库服务器执行的用于客户机的工作，工作通常包括处理要求访问数据库的查询。

正如任何服务，数据库服务可以进一步分类。用于数据库 **150** 的数据库服务可进一步分成 FIN 服务和 PAY 服务。FIN 服务是由数据库实例 **123** 和 **125** 执行的用于 FIN 应用程序的数据库服务。通常，该服务包括访问数据库 **150** 上的数据库对象，该数据库存储了用于 FIN 应用程序的数据库数据。PAY 服务是由数据库实例 **125** 和 **127** 执行的用于 PAY 应用程序的数据库服务。通常，该服务包括访问数据库 **150** 上的数据库对象，该数据库存储了用于 PAY 应用程序的数据库数据。

会话

为了使客户机与集群 **110** 上的数据库服务器交互，为该客户机建立会话。会话，例如数据库会话，是为客户机建立的至服务器（例如数据库实例）的特定连接，通过会话，客户机发出一系列请求（请求执行数据库语句）。对于数据库实例上建立的每个数据库会话，会话状态数据被保持为反映数据库会话的当前状态。这些信息包含了，例如，为其建立该会话的客户机的身份，和由在数据库会话内执行软件的进程产生的临时变量值。

客户机通过向集群 **110** 传送数据库连接请求建立数据库会话。接收器，如接收器 **190**，接收数据库连接请求。接收器 **190** 是运行在集群 **110** 上的一个进程，接收客户机数据库连接请求并把请求指

向集群 110 中的数据库实例。所接收到的客户机连接请求与服务(例如: 服务 FIN 和 PAY) 相关联。客户机请求被指向容留该服务的数据库实例, 并在此为客户机建立数据库会话。接收器 190 以对客户机透明的方式将请求指向特定的数据库实例和/或节点。接收器 190 可以运行在集群 110 中的任何节点上。一旦为客户机建立数据库会话, 客户机可以发布其它的请求, 可以以函数或远程过程调用的形式, 包括开始执行交易、执行查询、执行更新和其他类型的交易操作、提交或结束交易、以及结束数据库会话的请求。

监控工作负荷

资源被分配和再分配以满足资源上的性能级别和基数限制。为特定服务建立的性能级别和资源可用性在此被称作服务级别协定 (service-level agreement)。通常应用于多节点系统且并不一定应用于特定服务的资源上的性能级别和基数限制在此被称作策略。例如, 用于服务 FIN 的服务级别协定可能要求用于服务 FIN 的平均交易时间不超过给定阈值作为性能级别, 以及至少两个实例容留服务 FIN 作为可用性要求。策略可能要求任何节点的 CPU 利用率不应超过 80%。

在此策略还可以被称作后台策略, 因为它们被后台管理员用来管理整个系统的性能, 以及用来在认为资源不足以满足所有服务组的服务级别协定时, 在一组服务之间分配资源。例如, 策略为一个数据库指定相对于另一数据库较高的优先级。当资源不足以满足两个数据库的服务的服务级别协定时, 具有较高优先级的数据库, 以及使用该数据库的服务在分配资源时将会优先。

为了满足服务级别协定, 需要一种机制来监控和测量各种资源上的工作负荷。这些工作负荷的测量用于确定服务级别协定是否被满足, 以及根据满足服务级别协定所需来调节资源的分配。

根据本发明的实施例，工作负荷监视器容留在每个数据库实例上并且产生“性能度量”。性能度量是基于性能指标指示一个或多个资源或服务的性能等级的数据。在 Measuring Workload by Service (50277-2337)中描述了执行这些功能的方法。产生的信息可通过多节点数据库服务器 222 中的各种组件获得，多节点数据库服务器负责管理资源的分配来满足服务级别协定，这些会在后面详细描述。

可被用于测量指示性能或工作负荷的级别的特性或条件的特定类型的性能度量，在此被称作性能指标 (performance measure)。性能指标包括例如交易执行时间或 CPU 使用百分率。通常，涉及性能级别的服务级别协定可通过基于性能指标的阈值和标准来限定。

例如，交易的执行时间是性能指标。基于该指标的服务级别协定是，用于服务 FIN 的交易应该在 300 毫秒内执行。然而，另一个工作指标是节点的 CPU 使用百分率。基于该指标的后台策略是，节点不承受超过 80% 的使用率。

性能度量可指示集群的性能、在集群、集群中的节点、或特定数据库实例上运行的服务的性能。服务所特有的性能度量或指标在此被称作服务性能度量或指标。例如服务 FIN 的服务性能度量是为服务 FIN 执行的用于交易的交易时间。

根据实施例，服务级别协定和后台策略基于以下各项。

服务或数据库基数 服务级别协定或后台策略能够限制被分配给某个特定用途的资源数量。这种限制称为基数限制。服务基数限制要求最小和/或最大数目的数据库实例容留服务；数据库基数限制要求最小和/或最大数目的数据库实例在运行。基数限制也能限制集群中的节点数量。

服务或数据库优先级 一些服务或数据库可以被指定为具有高于其他的优先级。分配资源时优先考虑较高优先级的服务或数据库，特别当没有足够的资源满足所有的服务或数据库组的服务级别协定时。优先级信息可以由数据库管理员提供。

N_{cpu} 节点的 CPU 利用百分率。CPU 利用百分率可以通过对驻留在节点上的数据库实例上容留的服务的 CPU 利用百分率求和来计算。

X_{cpu} 节点的 CPU 利用率阈值形式的策略。为了说明的目的， X_{cpu} 对集群的所有节点是不变的。然而，它可能是变化的，取决于各种因素，例如为数据库运行的数据库实例的数量。

CPU_{aval} 这是节点上可用的未使用的 CPU 利用率，定义为 X_{cpu} 和 N_{cpu} 之间的差（即 $X_{cpu} - N_{cpu}$ ）。

可用性 可用性是基数的子类，因为它是特定用途可用的资源的最小数量。例如，服务可用性需求要求服务被至少最小数量的数据库实例容留。

可用性被称为可用性监视器的守护进程监视。可用性监视器，例如可用性监视器 192，是数据自适应鉴定监视器（daemon），检测何时节点或数据库实例由于不可用事件（例如系统崩溃）而变成不可用。当检测到节点或数据库实例变成不可用时，可用性监视器通知受影响的服务和节点的数据库导控器和/或集群导控器。导控器采取行动以满足可用性需求。例如，可用性需求要求 PAY 至少在两个数据库实例上容留。可用性监视器 192 检测到实例 125 已经停止并通知数据库导控器 152。作为响应，数据库导控器 152 启动服务扩展将服务 PAY 扩展到另一个运行的数据库实例，数据库实例 123。

导控器的层次

导控器的层次，如在 Hierarchical Management of the Dynamic Allocation of Resources in a Multi-Node System (50277-2382), 中描述的，用于在集群场 101 中动态调节资源的分配来满足服务级别协定。集群场 101 包括由集群场 101 上的数据库服务器管理的每个数据库的数据库导控器，集群场 101 中的每个集群的集群导控器，以及集群场 101 的场导控器。

数据库导控器，例如数据库导控器 152 和 162，动态地管理和调节由数据库的数据库实例容留的服务之间的数据库的资源分配。数据库导控器用来执行该职责的一种方法是如在 Incremental Run-Time Session Balancing in a Multi-Node System (50277-2411) 中描述的在数据库的数据库实例之间执行增量运行时会话平衡。增量运行时会话平衡在容留该服务的数据库实例间迁移服务的数据库会话。另一种数据库导控器可采取的方法是服务扩展。在服务扩展下，另一个数据库实例被分配以容留服务，如在 Service Placement for Enforcing Performance and Availability Levels in a Multi-Node System (50277-2412) 中所述的。

集群导控器，例如集群导控器 112，管理和调节数据库间的资源分配。集群导控器执行该职责采取的一种方法是向数据库向集群中的节点添加数据库实例或从其中删除数据库实例，如在这里详细描述。

场导控器，如场导控器 102，管理和调节集群间的资源分配。场导控器执行该职责采取的一种方法是向集群添加节点或从其中删除节点。

导控器检测服务级别协定的违反，在此称为服务级别违反。例如，数据库导控器 152 周期性分析性能度量并确定在实例 125 上的

用于FIN的平均交易时间违反了该指标的服务级别协定或确定节点124上用于服务FIN的CPU利用百分率违反了CPU利用的策略。

升级补救以解决服务级别协定的违反

根据实施例，通过在破坏性和成本更大的资源分配之前开始对破坏性和成本小的资源分配进行调节，数据库导控器补救它检测到的服务级别违反。在容留服务的数据库的数据库实例之间迁移服务的数据库会话通常比扩展服务到另一个数据库实例的破坏性和成本小；扩展服务到另一个数据库通常比提供数据库实例到另一个节点或添加另一个节点到集群的破坏性和成本要小。

图2示出了可以用来管理解决服务协议违反所采用的补救的升级的程序。这个程序由数据库导控器启动作为对检测到“源”数据库的“源”数据库实例上的服务的资源报警的响应。资源报警是触发对多节点计算机系统资源分配的关注条件或事件的检测。资源报警特别包括服务级别违反的检测。例如，资源报警可能检测到在源数据库实例125上的服务FIN的平均交易时间超过了该指标的服务级别协定。

参照图2，在步骤210，数据库导控器首先确定是否尝试运行时会话均衡。这个决定是通过确定在源数据库实例上是否有候选的服务用来执行会话均衡而做出的。如果确定应该执行会话均衡，则在步骤215会话均衡被执行，例如在Incremental Run-Time Session Balancing In A Multi-Node System (50277-2411)中所述的。

在步骤205，确定资源报警是否持续。如果在程序的迭代中调用的补救，例如在步骤215的会话均衡，解决了资源报警的根本原因，则资源报警可能不再持续。程序的执行结束。

然而，如果资源报警持续，另一个程序的迭代被执行以采取另一个补救。步骤 **205** 在每次迭代后被执行以判断是否迭代中采取的补救已经解决了资源报警或是否另一个程序的迭代应该被执行。

在步骤 **210**，如果确定运行时会话均衡不应该被尝试，则程序在步骤 **220** 确定是否应该尝试服务扩展。这个决定是通过确定在源数据库实例上是否有候选的服务可被扩展做出的，如在 *Service Placement for Enforcing Performance and Availability Levels in a Multi-Node System* (50277-2412) 中详细解释的。如果确定服务扩展应该被执行，则在步骤 **225**，服务扩展被执行，程序返回步骤 **205**。

如果确定服务扩展不能被执行，则数据库导控器确定资源报警的解决是否应该升级到集群导控器以实施更大成本的补救，例如为数据库分配另一个数据库实例，并且如果需要，向集群中添加节点。

在步骤 **230**，数据库导控器确定是否存在挂起的“NEED-INSTANCE”请求在由集群导控器为数据库导控器处理。NEED-INSTANCE 请求是请求另一个节点上的另一个数据库实例。挂起的 NEED-INSTANCE 请求是正在由例如集群导控器处理的 NEED-INSTANCE 请求。这种处理可以包括（如在后面详细描述）为数据库实例寻找节点，或从节点上删除数据库实例以便另一个数据库实例可以提供给该节点。如果存在挂起的 NEED-INSTANCE 请求，则在步骤 **235**，数据库导控器等待，然后执行步骤 **205**。如果资源报警持续，则数据库导控器执行接下来的迭代。作为挂起的 NEED-INSTANCE 请求的结果，新的数据库实例可能已经被分配给数据库。在下面的迭代中，数据库导控器可以扩展服务到新分配的数据库实例并减轻或补救资源报警。

如果在步骤 **230** 确定没有挂起的 NEED-INSTANCE 请求正在由集群导控器为数据库导控器处理，那么在步骤 **240**，数据库导控器

确定分配另一个数据库实例是否违反该数据库的最大基数。如果数据库导航器确定添加另一个数据库实例会违反数据库的最大基数，则执行进行到步骤 235，在此数据库导航器等待直到进行到可能执行图 2 所示的程序的另一个迭代。

如果在步骤 240，数据库导航器确定添加另一个数据库实例不会违反数据库的最大基数，则在步骤 245，数据库导航器通过传送 NEED-INSTANCE 请求将资源报警的解决升级到集群导航器。

当执行程序中的各步骤时，数据库导航器将各种动作指派给一个异步工作并等待由该工作执行的各种动作的结果。例如，数据库导航器可能正在等待其发布的扩展服务（见 *Service Placement for Enforcing Performance and Availability Levels in a Multi-Node System* (50277-2412)）或迁移会话（见 *Incremental Run-Time Session Balancing in a Multi-Node System* (50277-2411)）的异步工作的结果。另外，数据库导航器可以等待发布给集群导航器的 NEED-INSTANCE 请求的结果。当在这些情况中等待时，数据库导航器实际上没有闲置。相反，它同时执行图 2 所示的程序以处理其他资源报警。

当处理另一个服务的另一个资源报警时，数据库导航器在步骤 230 可以确定挂起的 NEED-INSTANCE 请求是挂起的。挂起的 NEED-INSTANCE 请求可能已经被数据库导航器为先前迭代中的另一个资源报警请求。

启动将数据库实例和节点分配给数据库

图 3 示出了响应于处理由“请求数据库导航器”发出的 NEED-INSTANCE 请求，接下来由集群导航器启动将节点分配给数据库。

参照图 3，在步骤 305，集群导航器确定在其负责的集群中是否有空闲的节点可以提供给实例。在实施例中，如果实例没有容留在节点上，并且集群导航器没有正在将该节点分配给另一个数据库实例（例如分配节点作为对为另一个数据库产生的 NEED-INSTANCE 请求的响应），或该节点被集群软件为某种目的而保留，则该节点是空闲的。集群导航器可以调用集群软件的 API 来确定哪些节点在该集群中以及数据库实例是否容留在其上或为某种目的已被集群软件保留。

在步骤 305，如果集群导航器确定节点为空闲，则在步骤 310，集群导航器调用集群软件的 API 函数在空闲的节点上为数据库启动数据库实例。接下来，集群导航器发送消息给请求的数据库导航器通知该数据库导航器这个数据库的实例已经提供给节点。

在步骤 305，如果集群导航器确定节点不是空闲，则集群导航器尝试获得集群中分配给其他数据库（“数据库牺牲者”）的节点。

在步骤 315，集群导航器定位数据库牺牲候选者。根据本发明的实施例，集群导航器产生嵌套在集群中的数据库的顺序列表，称作牺牲候选者列表，列表按照数据库优先级排序。具有较低优先级的数据库被定位在列表的开始处，具有较高优先级的数据库定位在列表的末端。

数据库牺牲候选者的定位可以通过调用函数来执行。函数根据函数中应用的策略返回一个或多个候选牺牲者。函数可以应用各种方法和策略来确定数据库牺牲候选者的列表。此外，这样的函数可以是用户提供的，即：函数可以不是数据库服务器的本地软件的部分，但已在数据库服务器注册，以为了确定数据库候选牺牲者的目的而被调用。

在步骤 320，集群导航器请求牺牲列表的数据库导航器以得到停止数据库实例的志愿者，以及从自愿的一个或多个数据库导航器中选择一个自愿者。

术语“停止”指的是禁止服务器的一组客户机对服务器的使用。停止数据库实例指的是禁止客户机使用这个数据库实例访问数据库。停止服务指的是禁止服务使用数据库实例。如详细描述，停止数据库实例可能需要阻止到数据库实例的新连接和将为客户机执行的工作通过例如会话迁移传送到另一个数据库实例。

步骤 320 可以以各种方式完成。

集群导航器可以按顺序一个一个地轮询牺牲候选者列表中的候选者的数据库导航器。为了轮询候选者的数据库导航器，集群导航器传送“QUIESCE-VOLUNTEER”请求给数据库导航器，数据库导航器通过发送指明数据库导航器是否自愿停止的消息来回应。如果轮询的候选者自愿，集群导航器选择该候选者作为牺牲者。

可选地，集群导航器将停止请求广播给所有的在牺牲候选者列表中的候选者的数据库导航器。然后，集群导航器在那些自愿的导航器（即发送指示该数据库导航器自愿的消息的导航器）中选择牺牲数据库。

由自愿的数据库导航器发送的消息可以包括指示停止成本的数据，将在后面详细解释。停止成本是停止数据库实例所需要的工作量。

牺牲数据库的选择可以基于数据库优先级，停止成本，或这些因素的组合。例如，集群导航器能选择与最低停止成本或最低数据库优先级相关联的数据库。如果基于数据库优先级选择，并且多个

数据库具有最低优先级，则集群导控器从多个数据库中选择与最低停止成本相关联的那一个。

在步骤 325，确定牺牲者是否被选择。如果没有牺牲者被选择，那么在步骤 330，牺牲数据库通过从具有至少两个数据库实例的候选牺牲者列表中选择最低优先级的数据库而被选出。执行进行到步骤 335。

在步骤 335，集群导控器发布异步工作。该工作向牺牲者的数据库导控器发送“QUIESCE-INITIATE”请求以请求数据库导控器停止数据库实例。

图 3 中执行的步骤结束。

接下来，集群导控器从牺牲者的数据库导控器接收“IDLE-INSTANCE”信息，作为响应集群导控器关闭停止的实例。同时，集群导控器处理其他请求，包括其他 NEED-INSTANCE 请求。

响应 Idle-Instance 消息

图 4 示出了根据本发明的实施例响应 IDLE-INSTANCE 请求，由集群导控器遵循的程序。在步骤 405，集群导控器接收 IDLE-INSTANCE 消息。IDLE-INSTANCE 消息包括被牺牲数据库导控器停止的“牺牲数据库实例”的身份和容留牺牲数据库实例的节点。

在步骤 410，集群导控器校验 IDLE-INSTANCE 消息的有效性，即校验牺牲数据库正在运行并且能被关闭。在牺牲数据库导控器已经停止牺牲数据库实例之后，牺牲数据库实例被关闭（即，使其不再运行），并且另一个数据库实例被启动以响应，例如，高可用性事件。在这种情况下，集群导控器不关闭另一个数据库实例。为了

确定在牺牲数据库实例停止的节点上运行的数据库实例的身份，集群导控器调用集群软件的 API。如果身份与牺牲数据库实例的身份相同，IDLE-INSTANCE 消息是有效的并且执行进行到步骤 415。否则，步骤的执行结束。

在步骤 415，集群导控器关闭牺牲数据库实例。在步骤 418，集群导控器确定等待新的节点的最高优先级数据库导控器（即其导控器正在等待新节点的最高优先级的数据库）。在步骤 420，集群导控器为最高优先级等待数据库启动请求的数据库实例，该数据库可能与最初向集群导控器提交 NEED-INSTANCE 消息的数据库不同。该程序允许后到的较高优先级请求取代可能已经启动实例的停止的较低优先级请求。步骤 415 和 420 都通过调用集群软件的 API 来执行。

响应 QUIESCE-VOLUNTEER 请求

图 5 示出了确定数据库导控器是否可能为数据库导控器的数据库自愿停止数据库实例的程序。该程序被执行来响应由“请求集群导控器”发送的“QUIESCE-VOLUNTEER”请求。

参照图 5，在步骤 505，数据库导控器通过为数据库少分配一个数据库实例来确定该数据库的最小数据库基数限制是否被违反。数据库导控器可能在停止另一个数据库实例的过程中。如果是这样，数据库导控器通过少分配一个数据库实例加上在被停止的过程中的数据库实例来确定最小数据库基数限制是否被违反。如果数据库导控器确定最小数据库基数限制被违反，则执行进行到步骤 530，在此数据库导控器发送消息给请求集群导控器，指定该数据库导控器不自愿停止数据库实例。否则，如果数据库导控器确定最小数据库基数限制不能被违反，则执行进行到步骤 510。

在步骤 510，数据库导控器通过为数据库分配少一个数据库实例来确定容留在数据库上的服务的最小服务基数是否被违反。不会违反容留在数据库上的任何服务的服务基数限制的数据库的实例的最小基数，是容留在数据库上的任何服务的最大的最小服务基数。例如，容留在数据库 150 上的数据库中，服务 FIN 和 PAY 的最小基数限制分别是 1 和 2。因此任何容留的服务的最大的最小基数限制是 2。存在用于数据库 150 的三个数据库实例。删除用于数据库 150 的一个数据库实例剩下 2，这样最小服务基数没有被违反，执行进行到步骤 515。如果，另一方面，服务 PAY 的最小基数限制是 3，那么任何服务的最大的最小服务基数是 3，删除用于数据库 150 的一个数据库实例违反容留在数据库 150 上的至少一个服务的最小服务基数，并且执行进行到步骤 530，在此数据库导控器给请求的集群导控器发送消息指示数据库导控器不自愿停止。

在步骤 515，为容留数据库的节点，即：容留数据库的数据库实例的节点，计算总的可用 CPU 利用率。根据实施例，这是通过对每个容留数据库的节点的 CPU_{aval} 求和来确定的。总的可用 CPU 利用率是容留数据库的节点组少一个节点时处理当前的工作负荷的能力的标志。例如，节点 122，124，和 126 的 CPU_{aval} 大于 80%。如果总的可用 CPU 利用率大于 X_{cpu} ，则有足够的总的可用 CPU 利用率在少一个节点时处理工作负荷而不违反阈值。

在步骤 520，确定是否有充足的总的可用 CPU 利用率可用。如果总的可用 CPU 利用率大于 X_{cpu} 加上 X_{cpu} 的因数，就有足够的总的可用 CPU 利用率可用。

计算的总的可用 CPU 利用率只是容留数据库的节点在少一个节点时处理当前的工作负荷而不违反服务级别协定和 CPU 利用率的策略的能力的估计。因为该估计可能不精确，向 X_{cpu} 添加因数以避免其余节点的过载。这个因数可以是，例如， X_{cpu} 的 5%。

如果,在步骤 **520**,确定没有足够的总的可用 CPU 利用率可用,则执行进行到步骤 **530**,这里数据库导控器发送消息指明其不自愿。否则,在步骤 **525**,数据库导控器发送消息确认其自愿。

确认数据库导控器自愿的消息也可以包含指示为数据库停止数据库实例的成本的数据。该数据包括任何数据库实例上的数据库会话的最小值,容留数据库的任何节点的最小 CPU 利用率百分率,和/或用于数据库的总的可用 CPU 利用率。通常,数据库实例上的数据库会话数越高说明停止实例的成本越高。较高的总的可用 CPU 利用率表示停止实例的成本较低。节点上较低的 CPU 利用率表示停止数据库实例的成本较低。

图 5 的过程的执行结束。

处理 Quiesce-Initiate 请求

图 6 示出了数据库导控器为数据库执行的从数据库的数据库实例中选择和停止数据库实例的程序。响应于由集群导控器发送的停止数据库实例的 QUIESCE-INITIATE 请求,数据库导控器执行该程序。

参照图 6,在步骤 **605**,数据库导控器选择数据库实例作为候选牺牲者,基于 CPU 利用率的顺序作出选择。具有较低的 CPU 利用率的数据库实例先于具有较高 CPU 利用率的数据库实例被选择。

在步骤 **610**,数据库导控器发送“BLOCK-INSTANCE”消息至候选牺牲者的接收器。消息指示接收器停止引导数据库连接请求至选定的候选者。因此,接收器将请求任何服务的数据库连接引导至容留该服务的另一数据库实例。

在步骤 **615**，数据库导控器确定数据库会话是否可迁移，即数据库会话是否能被迁移到另一数据库实例。例如，当会话状态包括一个打开文件的文件描述符时，数据库会话可能不可迁移。文件描述符包括只对源实例而不是目的实例上的会话有效的信息。其他确定数据库会话是否可以被迁移的方法在 Transparent Session Migration Across Servers (50277-2383) 中描述。如果会话是不可迁移的，“UNBLOCK-INSTANCE”消息在步骤 **630** 被发送给接收器，通知接收器引导连接请求至候选牺牲者。另一候选牺牲者在步骤 **605** 被选择。

如果会话能被迁移，执行进行到步骤 **620**。在该阶段，候选牺牲者成为牺牲者。

在步骤 **620**，会话从牺牲者迁移到数据库的另一数据库实例。会话可以在 Incremental Run-Time Session Balancing in a Multi-Node System (50277-2411) 中描述的相近的方式被迁移。如在此所述，会话被迁移到其他数据库实例来响应由服务级别违反导致的资源报警。服务的数据库实例上的会话被迁移到另一个数据库实例直到资源报警不再持续。

相似的程序可以被步骤 **620** 使用。数据库会话被迁移到数据库实例而不是牺牲者。资源报警，不是用于一个服务，而应用于牺牲者上的所有服务，并且持续直到所有的数据库会话从用于由牺牲者容留的所有服务的牺牲者迁出，或直到数据库导控器决定中止其迁移数据库会话的企图。在后一种情况，执行进行到步骤 **630**，在此数据库导控器发送 UNBLOCK-INSTANCE 消息给接收器。图 6 的程序被再次执行，除了牺牲者没有再次被选为牺牲者。最后，尽管从牺牲者迁出会话被执行，迁移会话至牺牲者的所有尝试被阻止。

在步骤 625, 数据库导控器向集群导控器发送 IDLE-INSTANCE 消息。

牺牲数据库的所有实例可能具有不可迁移的数据库会话。这种情况下, 数据库导控器选择一个实例作为牺牲者并迁移可迁移的数据库会话。对于不可迁移的数据库会话, 数据库导控器等待直到它们终止或成为可迁移的并且被迁移, 或它们可能基于管理员制定的策略和/或需要新节点的服务的优先级被强制地终止。

可选实施例的实例

通过在数据库服务和数据库服务的子类中动态分配多节点系统的资源, 描述本发明的实施例。然而, 本发明并不限于此。

例如, 本发明的实施例可用于在由应用程序服务器提供的服务中分配容留应用程序服务器的多节点系统的计算机资源。例如, 应用程序服务器是部分三层结构 (three tier architecture), 其中应用程序服务器位于客户机与数据库服务器之间。应用程序服务器主要用于存储应用程序代码、提供对应用程序代码的访问, 以及执行应用程序代码, 而数据库服务器主要用于存储和提供对用于应用程序服务器的数据库的访问。应用程序服务器将对数据的请求发送至数据库服务器。响应于执行存储于应用程序服务器上的应用程序代码, 该请求可由应用程序服务器产生。应用程序服务器的一个实例是 Oracle 9i Application Server 或 Oracle 10g Application Server。类似于在此所述的多节点服务器的实例, 应用程序服务器可分布为在多个节点上执行的多个服务器实例、容留可在服务器实例之间迁移的多个会话的服务器实例。

本发明也不限于仅由执行相同软件产品或相同版本的软件产品的拷贝的服务器实例所构成的同质的多节点服务器。例如, 多节

点数据库服务器可由多组服务器实例构成，每组执行来自不同厂商的不同的数据库服务器软件，或执行来自相同厂商的不同版本的数据库服务器软件。

硬件综述

图 7 是示出可以执行本发明的实施例的计算机系统 700 的框图。计算机系统 700 包括用于传递信息的总线 702 或其它通信装置以及用于处理信息的与总线 702 连接的处理器 704。计算机系统 700 还包括连接至总线 702 的主存储器 706，诸如随机访问存储器 (RAM) 或者其它动态存储装置，用于储存信息和将由处理器 704 执行的指令。在执行将由处理器 704 执行的指令期间，主存储器 706 还可用于储存临时变量或其他中间信息。计算机系统 700 进一步包括只读存储器 (ROM) 708 或连接至总线 702 的其他静态存储装置，用于存储静态信息和处理器 704 的指令。提供诸如磁盘或光盘的存储设备 710，并连接至总线 702 用于存储信息和指令。

计算机系统 700 可以经由总线 702 连接至诸如阴极射线管 (CRT) 的显示器 712，用于向计算机用户显示信息。包括字母数字键和其他键的输入装置 714 连接至总线 702，用于将信息和指令选择传递到处理器 704。另一种类型的用户输入装置是光标控制 716，诸如鼠标、跟踪球、或光标方向键，用于将方向信息和命令选择传递到处理器 704 并用于控制显示器 712 上的光标移动。输入装置通常在两个轴上 (第一个轴 (例如 X 轴) 和第二个轴 (例如 Y 轴)) 具有两个自由度，使装置能指派平面上的位置。

本发明涉及计算机系统 700 的使用，用于执行在此描述的技术。根据本发明的一个实施例，通过计算机系统 700 响应于执行包括在主存储器 706 中的一个或多个指令的一个或多个序列的处理器 704，来实现这些技术。这样的指令可以从诸如存储装置 710 的其它计算

机可读介质读入主存储器 706。包括在主存储器 706 中的指令序列的执行，使得处理器 704 执行此处所述的处理步骤。在可选实施例中，可以使用硬连线电路（hard-wired circuitry）来取代软件指令或者与软件指令结合来实施该发明。因此，本发明的实施例将不限于硬件电路和软件的任何特定组合。

这里使用的术语“计算机可读介质”是指参与向处理器 704 提供指令用于执行的任何介质。这种介质可以采取多种形式，包括但不限于非易失性介质、易失性介质、和传递介质。非易失性介质举例来说包括光盘或磁盘，诸如存储装置 710。易失性介质包括动态存储器，诸如主存储器 706。传输介质包括同轴电缆、铜线、和光纤，包括组成总线 702 的导线。传输介质还可采取声波或光波形式，例如那些在无线电波和红外线数据通信过程中产生的声波和光波。

通常形式的计算机可读介质包括如软盘、软性盘、硬盘、磁带，或者任何其它磁性介质、CD-ROM、任何其它光介质、打孔纸、纸带、或者任何带孔图样的物理介质、RAM、PROM、EPROM、FLASH-EPROM、或者其他任何存储芯片或者盒式磁带，或者以下提到的载波、或者计算机可读的任何其他介质。

各种形式的计算机可读介质可参与将一个或者多个指令的一个或多个序列承载到处理器 704 用于执行。例如，指令开始可承载在远程计算机的磁盘中。远程计算机可以将指令加载到其动态存储器中，然后使用调制解调器通过电话线发送指令。计算机系统 700 本地的调制解调器可接收电话线上的数据，并使用红外发射器将数据转换成红外信号。红外探测器可以接收红外信号携带的数据，并且合适的电路可以将数据放到总线 702 上。总线 702 将数据承载到主存储器 706，处理器 704 从主存储器取回并执行这些指令。在由处理器 704 执行这些指令之前或之后，由主存储器 706 接收的指令可随意地储存在存储装置 710 上。

计算机系统 **700** 还包括连接至总线 **702** 的通信接口 **718**。提供双向数据通信的通信接口 **718**，连接到与局域网 **722** 连接的网络链路 **720**。例如，通信接口 **718** 可以是综合业务数字网 (ISDN) 卡或者调制解调器，用于提供到相应类型的电话线的数据通信连接。又如，通信接口 **718** 可以是局域网 (LAN) 卡，用于提供至兼容局域网 (LAN) 的数据通信连接。也可以使用无线链路。在任何这样的实施中，通信接口 **718** 发送和接收承载表示各种类型的信息的数字数据流的电信号、电磁信号、和光学信号。

网络链路 **720** 通常可通过一个或者多个网络向其它数据装置提供数据通信。例如，网络链路 **720** 可通过局域网 **722** 与主机 **724** 连接，或者与互联网业务提供商 (ISP) **726** 操作的数据设备连接。ISP **726** 又通过目前通称为“互联网” **728** 的全球分组数据通信网络提供数据通信业务。局域网 **722** 和互联网 **728** 都使用承载数字数据流的电信号、电磁信号、或光学信号。通过各种网络的信号和网络链路 **720** 上的信号以及通过通信接口 **718** 的信号，都传送数字数据给计算机系统 **700** 或者传送来自计算机系统的数字数据，是传输信息的载波的示例性形式。

计算机系统 **700** 能通过网络、网络链路 **720**、和通信接口 **718** 发送消息和接收数据 (包括程序代码)。在互联网的实例中，服务器 **730** 可通过互联网 **728**、ISP **726**、局域网 **722**、和通信接口 **718**，传送用于应用程序的所请求的程序代码。

所接收的代码可以在其被接收时由处理器 **704** 执行，和/或储存在存储装置 **710** 或者其它非易失性介质中用于随后执行。按照这种方式，计算机系统 **700** 可以以载波的形式获得应用代码。

以上所述仅为本发明的优选实施例而已，并不用于限制本发明，对于本领域的技术人员来说，本发明可以有各种更改和变化。凡在本发明的精神和原则之内，所作的任何修改、等同替换、改进等，均应包含在本发明的保护范围之内。

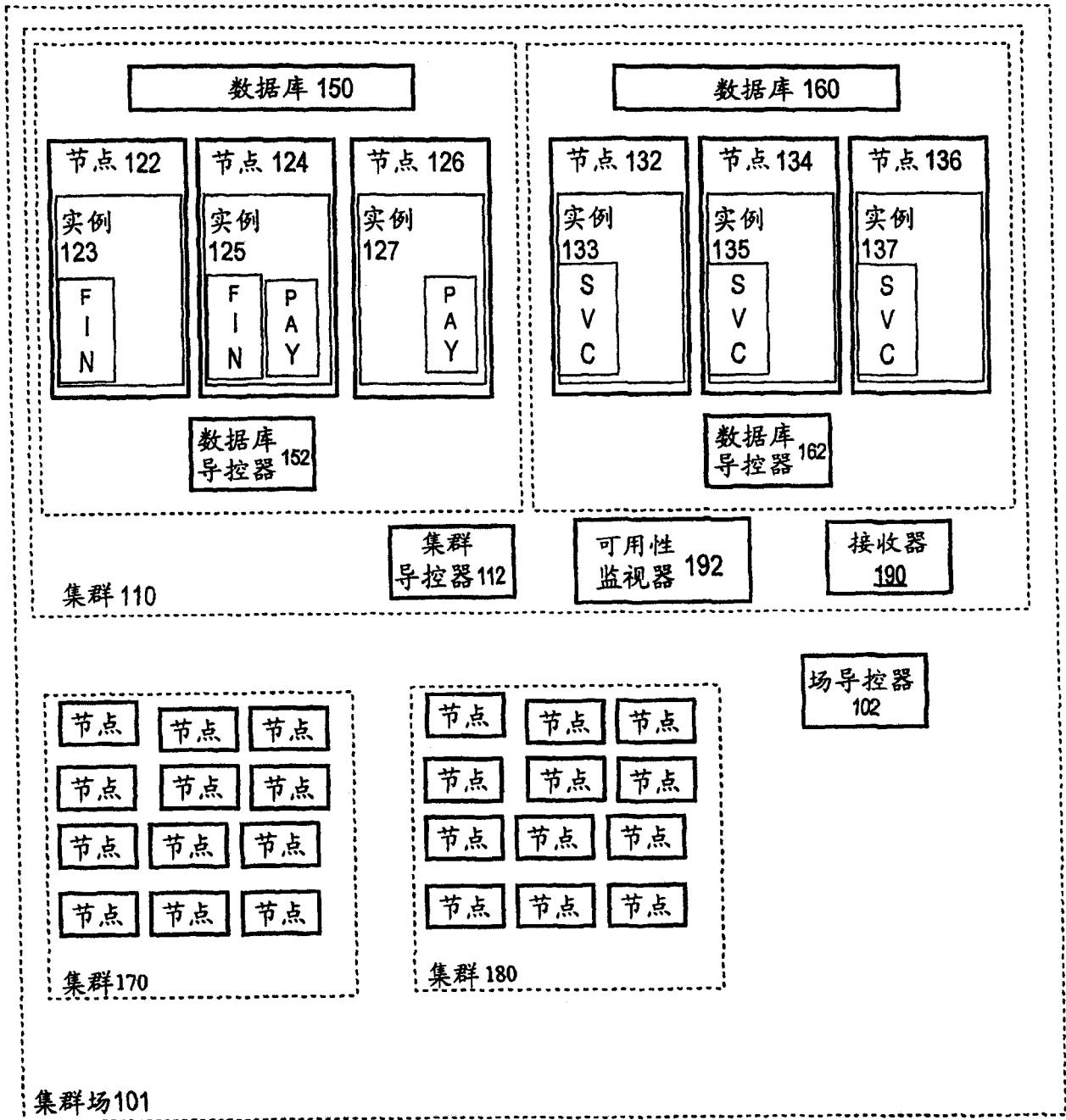


图 1

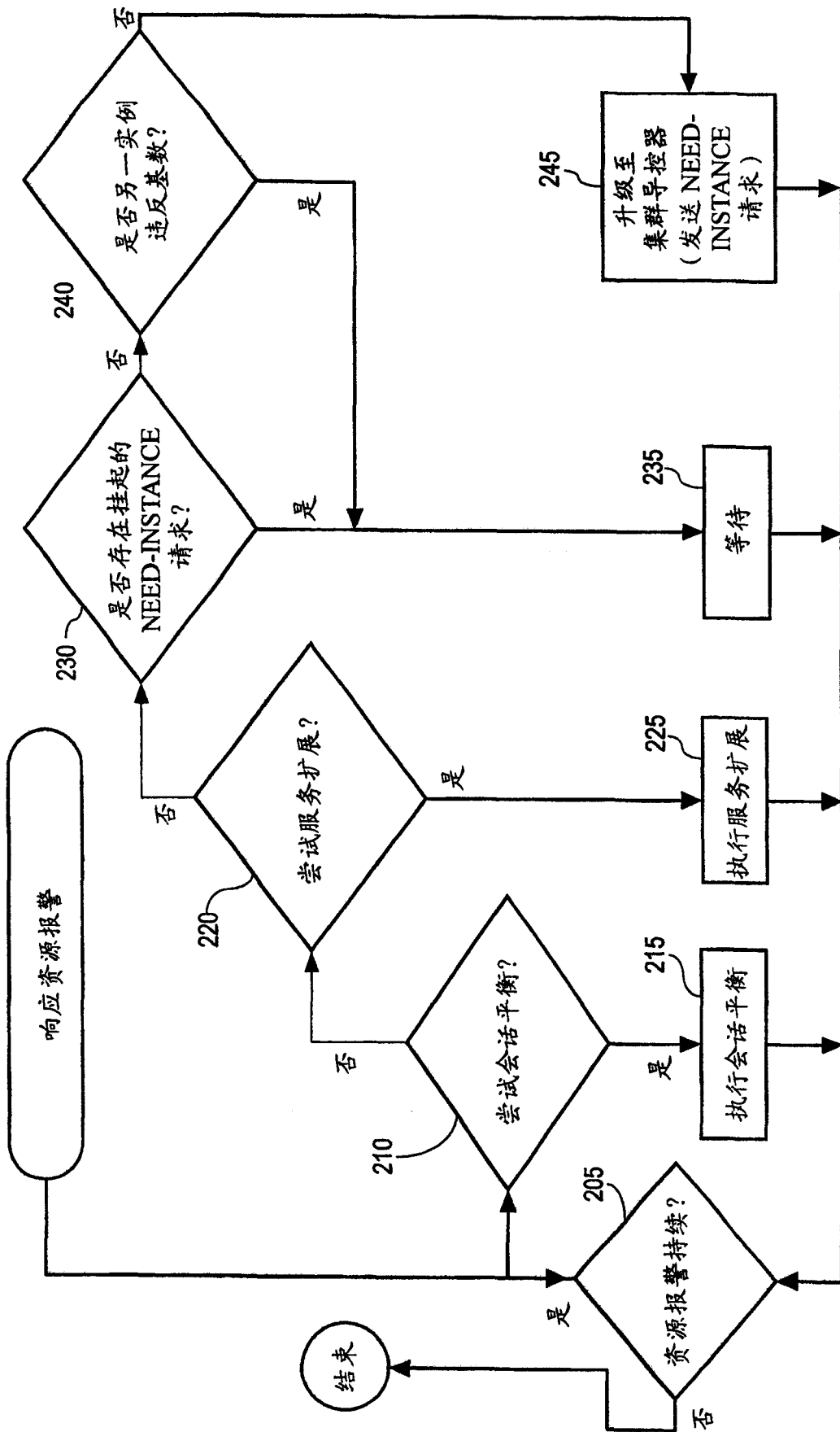


图 2

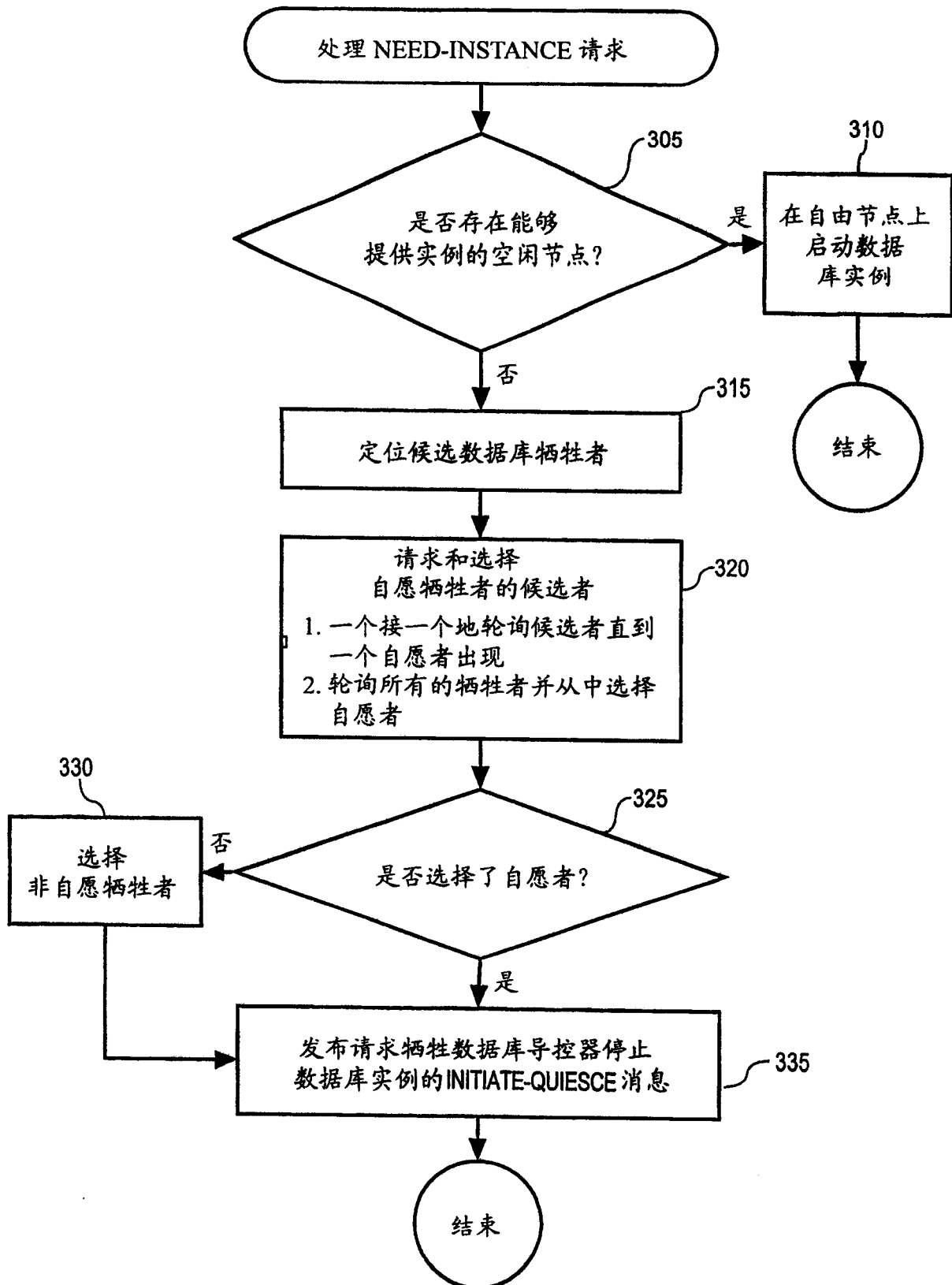


图 3

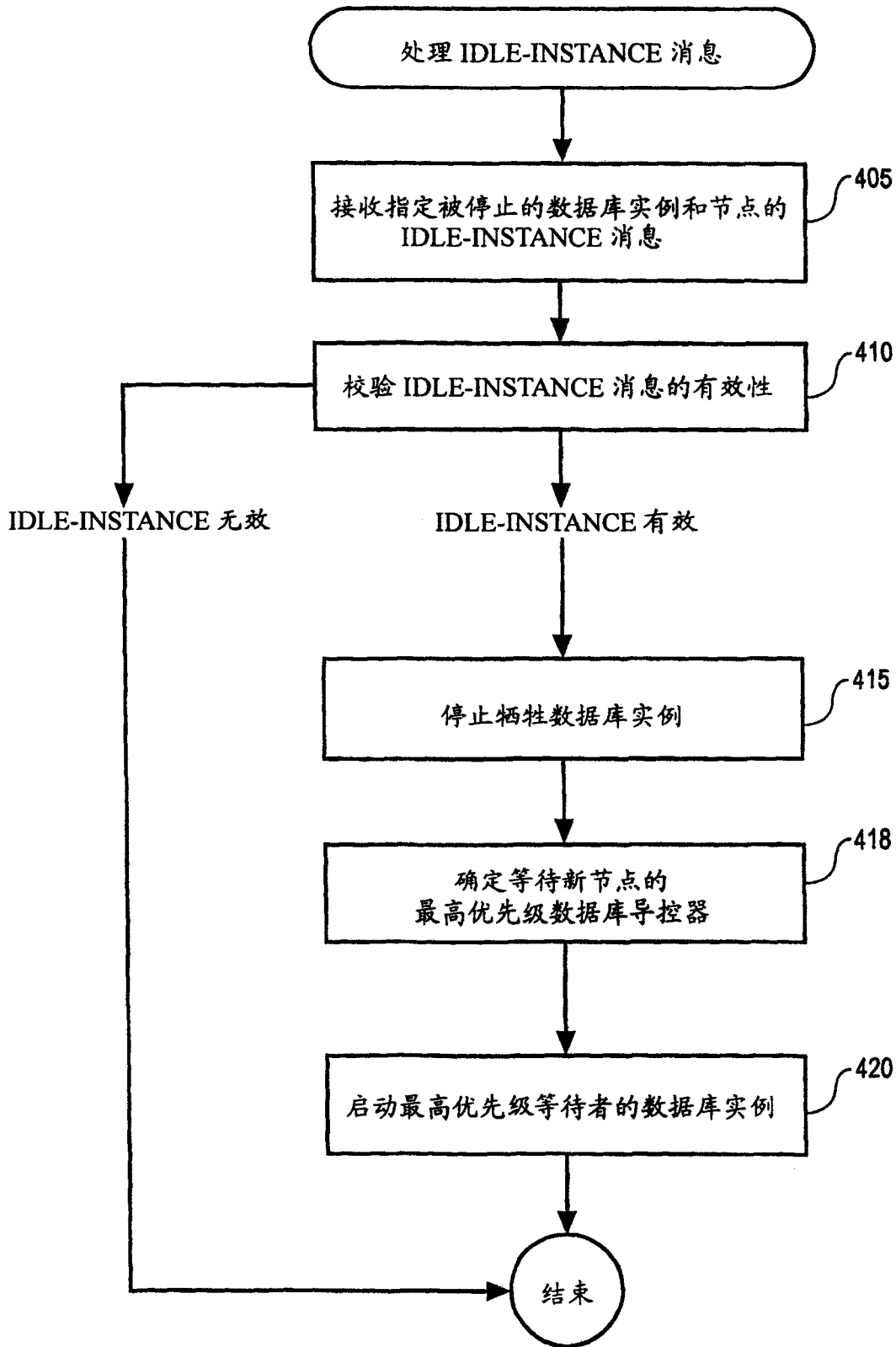


图 4

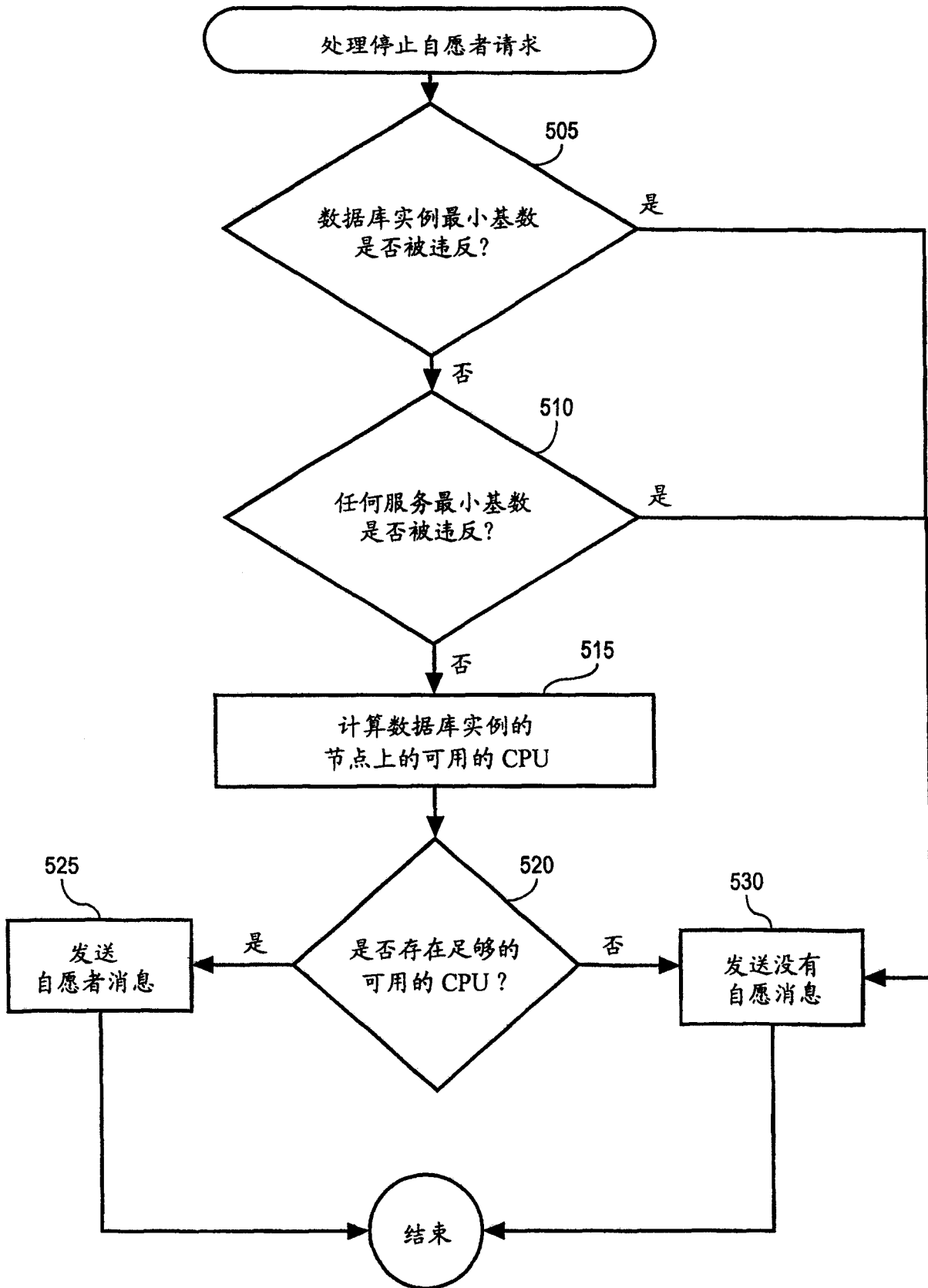


图 5

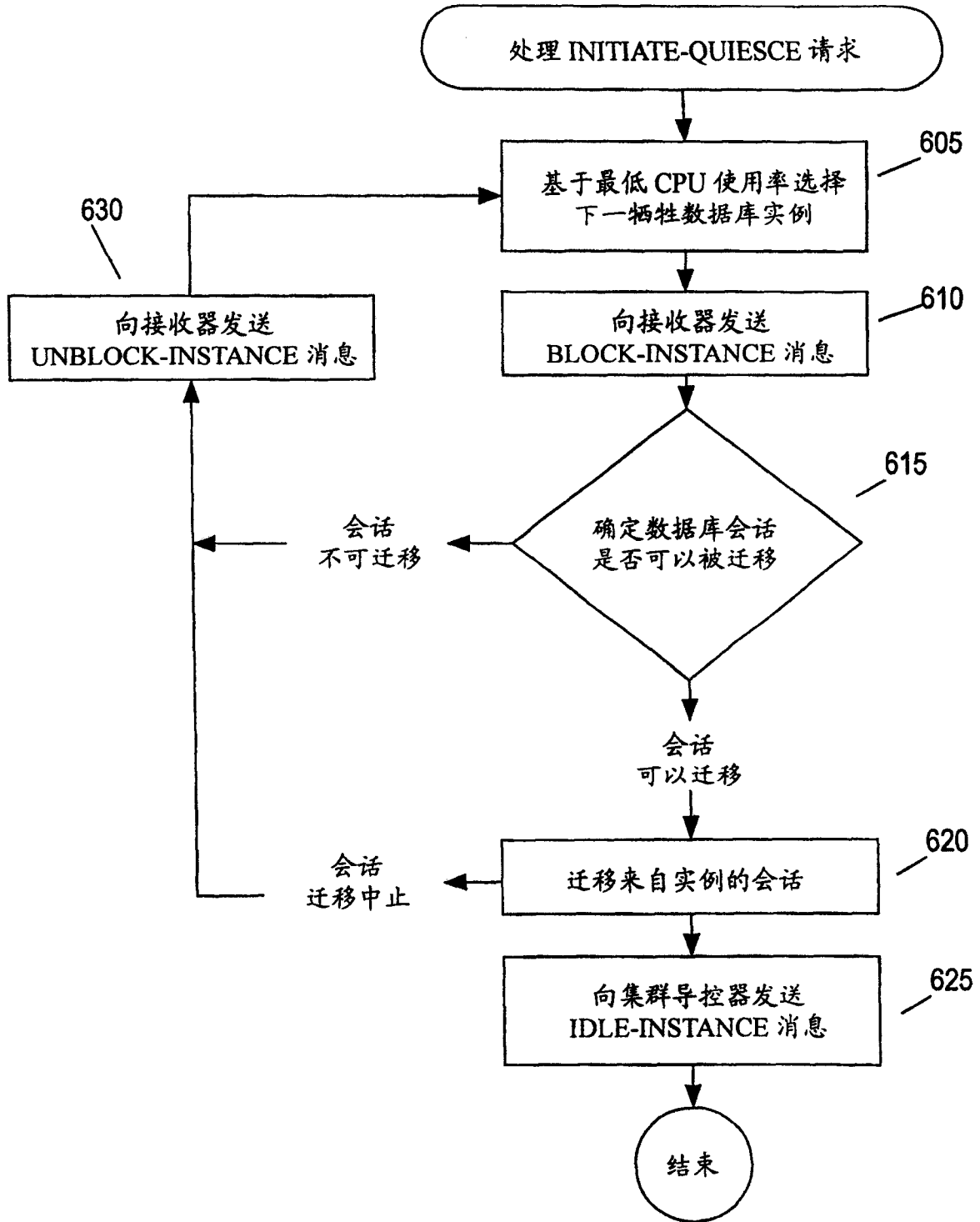


图 6

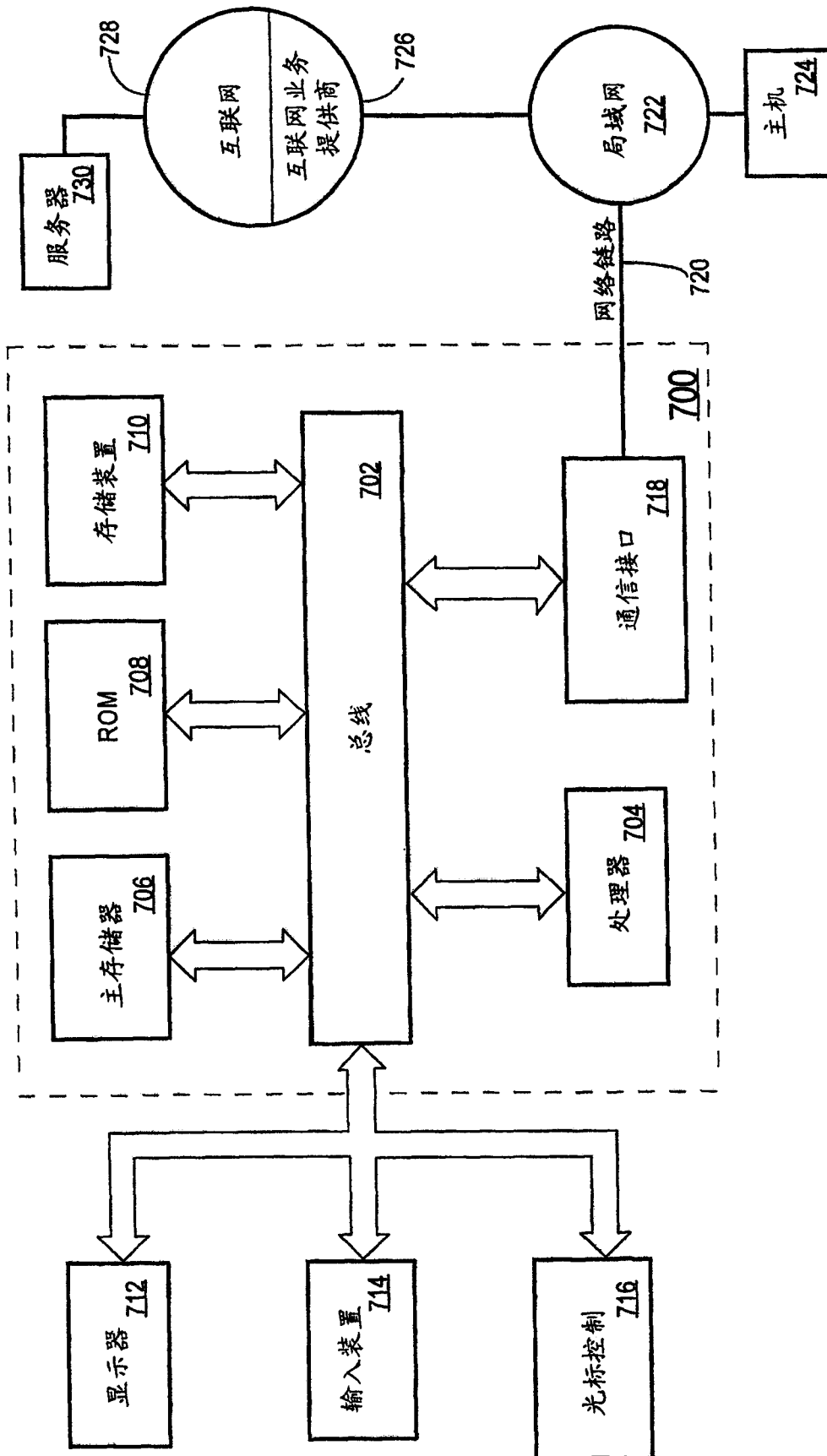


图 7