

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 17/30 (2006.01)



[12] 发明专利说明书

专利号 ZL 02824692.6

[45] 授权公告日 2010 年 1 月 6 日

[11] 授权公告号 CN 100578490C

[22] 申请日 2002.10.21 [21] 申请号 02824692.6

[30] 优先权

[32] 2001.10.25 [33] US [31] 60/335,633

[32] 2002.8.5 [33] US [31] 10/212,382

[86] 国际申请 PCT/US2002/033659 2002.10.21

[87] 国际公布 WO2003/036517 英 2003.5.1

[85] 进入国家阶段日期 2004.6.10

[73] 专利权人 BEA 系统公司

地址 美国加利福尼亚州

[72] 发明人 迪安·B·雅各布斯 罗布·伍伦
塞思·怀特

[56] 参考文献

US5546579A 1996.8.13

US6298478B1 2001.10.12

US5999947A 1999.12.7

US5897634A 1999.4.27

审查员 夏彩杰

[74] 专利代理机构 北京市柳沈律师事务所

代理人 邵亚丽 马莹

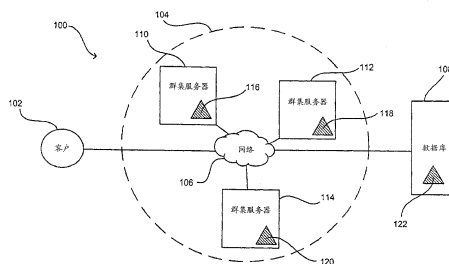
权利要求书 3 页 说明书 8 页 附图 6 页

[54] 发明名称

刷新 bean 缓存的系统和方法

[57] 摘要

网络群集(104)中的服务器(110、112、114)可分别存储数据项(122)的副本(116、118、120)在本地缓存中,通过只读实体 bean 提供对这些副本的读访问。数据库(108)中的原始数据项(122)可通过其中一个群集服务器上的读/写实体 bean 来更新。群集服务器可访问使无效目标,它包含有群集中服务器上存储的数据项的副本(116、118、120)相关的标识信息。一旦读/写 bean 更新了数据库中的数据项,使无效请求可发送或多点传送到所有群集成员,或者到使无效目标上所包含的任何只读 bean 或者服务器。每个接收请求的服务器或只读 bean 知道在本地缓存中废弃该数据项的任何副本,并且可从数据库请求该数据项的当前副本。



1. 一种用于在数据库中更新数据项的缓冲副本的方法，包括：
在服务器上存储数据项的副本，该数据项存储在数据库中；
产生只读 bean 来提供对副本的读访问以及存储与只读 bean 相关的标识信息；
产生读/写 bean 来提供到数据库中的数据项的读和写访问，以及使用读/写 bean 来更新数据项；以及
使用标识信息来发送使无效请求到只读 bean 以使数据项的副本可被删除。
2. 如权利要求 1 所述的方法，其中：
存储数据项的副本的步骤把副本存储在第一服务器上的缓存中。
3. 如权利要求 2 所述的方法，其中：
产生只读 bean 的步骤在第一服务器上产生只读 bean；和
产生读/写 bean 的步骤在第二服务器上产生读/写 bean。
4. 如权利要求 3 所述的方法，其中：
发送使无效请求的步骤从包含读/写 bean 的第二服务器发送请求到包含只读 bean 的第一服务器。
5. 如权利要求 1 所述的方法，其中：
产生只读 bean 和存储标识信息的步骤存储标识信息为 XML 文档中的一个标记。
6. 如权利要求 1 所述的方法，其中：
发送使无效请求到只读 bean 的步骤进一步包括多点传送请求到任何只读 bean，该只读 bean 存储作为由标识信息所标识的数据项的副本。
7. 如权利要求 1 所述的方法，其中：
发送使无效请求到只读 bean 的步骤进一步包括多点传送请求到群集的任何成员，服务器是群集的成员。
8. 如权利要求 7 所述的方法，进一步包括：
周期性经过群集多点传送使无效请求。
9. 如权利要求 1 所述的方法，其中：
发送使无效请求到只读 bean 的步骤通过点到点连接发生。

10. 如权利要求 1 所述的方法, 其中:

发送使无效请求的步骤包括多点传送包含版本号的请求。

11. 如权利要求 1 所述的方法, 进一步包括:

在使无效请求中包括版本标识符。

12. 如权利要求 11 所述的方法, 进一步包括:

通过检查版本标识符来确定对于只读 bean 的使无效请求是否丢失; 以及
如果只读 bean 丢失了使无效请求, 则请求向只读 bean 重发使无效消息。

13. 如权利要求 1 所述的方法, 进一步包括:

在发送使无效请求后, 存储最近的使无效请求。

14. 如权利要求 1 所述的方法, 进一步包括:

在更新数据项之后、并且在发送使无效请求之前, 收集多个更新的信息
并且批处理更新信息到一个使无效请求中。

15. 一种用于在群集中更新数据项的缓冲副本的系统, 包括:

在所述群集中的第一服务器, 其上缓冲所述数据项的第一副本和允许对
所缓冲的第一副本的读访问的第一只读 bean;

在所述群集中的第二服务器, 其上缓冲所述数据项的第二副本和允许对
第二缓冲副本的读访问的第二只读 bean;

其中,

在所述群集中的一服务器上缓冲允许对所述数据项的写访问的读/写
bean, 该读/写 bean 被编程为当所述读/写 bean 更新所述数据项时发送使无
效请求到第一和第二只读 bean。

16. 如权利要求 15 所述的系统, 进一步包括:

适合于包含数据项的数据库。

17. 如权利要求 15 所述的系统, 其中:

第一只读 bean 有允许删除数据项的第一缓冲副本的接口; 和

第二只读 bean 有允许删除数据项的第二缓冲副本的接口。

18. 如权利要求 15 所述的系统, 其中:

如果没有接收到请求时, 第一和第二只读 bean 的每一个都适合于请求重
新发送使无效请求。

19. 如权利要求 18 所述的系统, 其中:

如果无法获得使无效请求的重新发送时, 第一和第二只读 bean 的每一个

都适合于读取数据项。

20. 如权利要求 15 所述的系统, 进一步包括:

包含每一个只读 bean 的标识符的使无效目标, 该只读 bean 在缓冲数据项的副本的群集中。

21. 如权利要求 20 所述的系统, 其中:

使无效目标进一步包含读/写 bean 的标识符。

22. 如权利要求 20 所述的系统, 其中:

为了发送使无效请求到缓冲数据项的副本的每一个只读 bean, 读/写 bean 适合于检查使无效目标。

23. 如权利要求 15 所述的系统, 其中:

在处理使无效请求之后, 第一和第二只读 bean 中的每一个还适合于读取数据项的新副本。

刷新 bean 缓存的系统和方法

优先权要求

本申请要求下列申请的优先权，并在此引入作为参考：

美国临时专利申请 No.60/335,633，由 Dean Bernard Jacobs 和 Rob Woollen 在 2001 年 10 月 25 日提交，题目为“SYSTEM AND METHOD FOR FLUSHING BEAN CACHE”。

美国专利申请 No.10/212,382，由 Dean Bernard Jacobs、Rob Woollen 和 Seth White 在 2002 年 8 月 5 日提交，题目为“SYSTEM AND METHOD FOR FLUSHING BEAN CACHE”。

版权通告

本专利申请所公开内容的一部分包含有受版权保护的材料。版权所有人不反对任何人对专利文档或专利公开内容进行如专利和商标办公室专利文件和记录中所述的翻印复制，但除此之外将保留所有的各种版权权利。

参照引用案例

下列申请被参照引用并在此作为引用作为参考：

美国临时专利申请 No.60/305,986，由 Dean Bernard Jacobs、Reto Kramer 和 Ananthan Bala Srinivasan 在 2001 年 7 月 16 日提交，题目为“DATA REPLICATION PROTOCOL”。

美国临时专利申请 No.60/316,187，由 Dean Bernard Jacobs 和 Rob Woollen 在 2001 年 8 月 30 日提交，题目为“CLUSTER CACHING WITH CONCURRENCY CHECKING”。

技术领域

一般而言，本发明涉及在网络中存储数据的一种系统和方法。

背景技术

当数据项存储在可通过网络访问的单个数据库或数据存储器中时，通常情况下，多个服务器或客户将需要访问该数据项。传统而言，这需要在每次访问数据项时都存取一次数据库。每次都存取数据库是资源相对密集(resource intensive)和效率相对低下的。

克服一部分效率问题和伸缩级性问题的一种方法是在缓存中存储数据项的本地副本。当将来需要访问该数据项时，服务器或客户则可以使用此本地副本。这个做法在数据项永远不变时是适用和高效的，但当数据项在数据库中被更新时，问题出现了。

如果数据库中的数据项被更新了，存储在网络上的本地缓存中数据项的副本将不同于数据库中的数据项，因为缓存不会自动接收到更新。当在网络上的多个服务器和/或客户上存在本地副本时，问题更严重了。因为这些本地副本的每一个都在不同时刻被创建，在网络上可存在该数据项的多个版本。如果用户试图更新或察看该数据项，该用户所访问的副本可能不是当前的和正确的。

这种数据等待时间(latency)问题对于需要接近实时精确性的应用可能导致严重的问题，例如提供“实时”股价的网站。这种应用可能会用到有至少两个列的数据表，一个列包含股票符号，可作为表的主关键字，和一个列包含每种股票的当前价格。在这种应用中，多数活动包括用户访问网站和读取当前股市价值。同样典型的活动还包括后台应用或系统周期性访问，例如每分钟一次，带来更新的股价。为了更新数据，这些后台系统需要读/写访问数据库。

对系统的多数访问是只读的。对于这些只读用户，系统可缓冲数据来提供更快的访问。系统可周期性更新缓冲信息，例如每十五分钟。然而，在这种“通常为读”的情况下，为用户提供最近的数据可能会更好。在提供准确信息上，十五分钟的延迟对于许多应用来说可能是不合要求的。通常最好能为用户提供尽可能准确的信息。

确保用户获得准确信息或至少与存储在数据库中的数据一致的信息的一种方法是，对于每个请求，从数据库中拉取信息，而不是读取缓冲副本。这对于许多应用来说可能是非常昂贵的，因为访问数据库和从内存中读取值相比，需要更大量的时间和更资源密集。

对于在数据库中更新数据的人来说，把尽可能多的更新都包括在一批事

务中以改善性能可能是较理想的。把更新都包围在单次事务中也确保所有更新都发生或者任何更新都没有发生。但是，问题出现在，如何为在事务中更新的每一项更新其缓冲副本。

发明内容

这里包括了一种系统和方法，用于在网络群集中至少一个服务器上更新存储在本地缓存中的数据项副本。标识(Identification)信息将提供给存储在群集中服务器上的读/写 bean。标识信息涉及在群集中包含只读 bean 和本地缓存中该数据项副本的任何服务器。只读 bean 提供该数据项的本地副本的读访问。原始的数据项存储在网络数据库中，使用读/写 bean 来更新。当数据项由读/写 bean 更新时，使无效(invalidate)的请求可从包含读/写 bean 的服务器发送或多点传送到整个群集，或发送到标识信息所识别的有该数据项的本地副本的任何服务器或只读 bean。响应于此请求，于是该数据项的任何本地副本可被废弃。该数据项的当前副本可从数据库读取和存储在本地缓存中。

附图说明

图 1 示出本发明的一个实施例所述的系统。

图 2 示出如图 1 的实施例所述的系统。

图 3 示出图 1 所示系统的另一个实施例。

图 4 示出如本发明的一个实施例所述的方法的流程图。

图 5 示出如本发明的另一个实施例所述的方法的流程图。

图 6 示出如本发明的另一个实施例所述的系统。

具体实施方式

为了在网络上分布的项之间维护一致性，如本发明所述的系统可利用 bean 或 Java Beans。Bean 基本上是由于可加到服务器上扩展功能的组件的框架。一个实施例使用两种 bean，“只读”实体 bean 和“读/写”实体 bean。实体 bean 是持久化、允许共享访问、有主关键字并且可与其他实体 bean 相互共享。每个实体 bean 可在关系数据库中有基础表，bean 的每个实例对应该表的一行。

只读 bean 是可在服务器上缓冲的 bean，例如驻留在网络群集中的企业级

Java Bean。只读 bean 可提供到群集中的任何服务器和群集内外的任何客户的读访问，读/写 bean 是事务性的，驻留在群集服务器上，为群集服务器提供到网络数据库的读/写访问。只读 bean 处理群集服务器上本地缓存中的数据。读/写 bean 处理数据库中的信息。

处理缓存和数据库中的信息并发的一种方法是为每个只读实体 bean 联结一个超时值。例如，只读 bean 可配置为十分钟的缺省循环。在经过十分钟的周期之后，只读 bean 重回数据库和读取当前值。这个办法在某些应用中很好，例如以固定间隔改变值的那些应用。

但是，也有数据很少发生变化的应用。当数据改变时，用户可能需要尽快知道变化。因为数据不常变化，为了节约资源而设置长的读循环时间是很吸引人的。但是，这可能会有导致与数据有关的等待时间的非期望结果，因为取决于在周期中更新发生的时机，在更新数据时的等待时间结果可能差不多会和周期时间一样长。对于这些应用，最好是在更新数据库中的数据后立即更新只读用户可访问的数据。

如本发明所述的一种系统提供了只读 bean 所暴露的接口。此接口在用户更新数据项或得知更新时允许用户或应用程序告诉系统来废弃缓存，或“使缓存无效”。此接口应由“CachingHome”引用，因为实体 bean 通常都有创建它的“Home”或 factory。CachingHome 可在其上有三个方法，和可如下编码：

```
package weblogic.企业级 Java Bean;

public interface CachingHome {

    public void invalidate(Object pk) throws RemoteException;
    public void invalidate(Collection pks) throws RemoteException;
    public void invalidateAll() throws RemoteException;
}
```

方法 invalidate(Object pk)允许用户使与数据库或数据表中的特定主关键字相关联的数据无效。方法 invalidate(Collection pks)允许用户使关键字的集合或组对应的数据无效。方法 invalidateAll()允许用户使数据表中所有关键字对应的数据无效。这些使无效方法允许用户确保那些值存储在本地缓存中，直到程序员、应用程序、用户或系统另有指示。

这样的方法 300 如图 5 的流程图中所示。数据项的副本存储在网络群集中的至少一个服务器上 302。可使用位于群集服务器其中一个上的读/写 bean 在数据块中更新数据项 304。然后可使用位于其中一个服务器上的只读 bean 的接口来发起使无效请求，将请求发送到包含数据库的本地副本的任何服务器上 306。接收到请求的服务器上该数据项的任何副本则可废弃了 308。

在具有网络群集 104 的系统 100 中，例如图 1 所示，很可能在群集 104 中的每一个服务器 110、112、114 上都缓冲了存储在数据库 108 中的值 122 的副本。如果客户 102 通过网络 106 与服务器 110 联络，请求该服务器 110 使指定关键字无效，例如发出请求 “invalidate(Key)”，服务器 110 要废弃值 122 的缓冲副本 116 或与该关键字关联的值是很容易的。但是，存在的问题是，如何让服务器 112 和 114 也废弃它们的缓存副本 118 和 120。

一个实施例允许服务器 110 当从客户 102 接收到使无效请求 124 时废弃本地缓存中的副本 116，如图 2 所示。在从本地内存中废弃副本 116 后，服务器 110 可用多点传送发送消息 126 到其他服务器 118、120 或群集 104 中的只读 bean 来废弃本地缓存中该值的副本。多点传送是在不需要任何不必要的包复制的条件下来从一个服务器或源发送信息包或消息到其他服务器的技术，其原因是一个包是从来源发送和在网络中根据需要复制的。这种办法允许每个服务器当数据库中的值更新时废弃缓冲值。

图 6 展示了包括多个客户 402 和多个数据库 416 通过网络 404 通信的另一个系统。数据项 418 存储在数据库 416 的其中一个里。该数据项的副本 420 存储在群集 408 的群集服务器 406 中。有两个副本 422 存储在群集 412 的群集服务器 422 中，并且服务器 414 上的副本 424，并不包含在任何服务器群集中。此系统类似于图 1 和图 2 的系统那样工作，其原因是包含数据项 418 的副本的服务器 406、410、414 的其中一个当从客户接收到使无效请求时废弃本地缓存中的副本，并且可用多点传送发送消息到网络上的其他服务器来废弃本地缓存中该值的任何副本。

由于事实上来源只发送一次多点传送消息并且不等待其他服务器的接收确认，存在另一个问题。群集中的服务器可能没获得使无效请求，例如是否它暂时离线。如本发明所述的系统可通过使用版本号或顺序号标记每一个这样的消息或请求来提供更可靠的多点传送。这样，接收请求的服务器将知道请求的版本，以及它接收的上个请求的版本，因此服务器将知道它是否丢失

了某个消息。一旦服务器确定它已丢失了一个消息，它可以请求重发消息以便相应地更新。

但是，这种办法的一个问题是，服务器只有在另一个更新发送后才能知道它是否已丢失了一个更新。在特定应用例如为周 1、2 和 3 发布每周特供的在线商店中，只有等到下次更新才能获取正确信息是不可接受的。商店只有在周 3 的更新时才发现它已丢失了周 2 的特供。周 1 的特供可能在周 2 一直存在，同时显示错误信息给那时访问系统的任何用户。当系统发现它丢失周 2 的更新时，已经是周 3 了。在没有曾经显示给用户的信息的情况下，服务器将以简单地抛弃周 2 的信息而告终。

如本发明所示的系统可通过周期性“如心跳般地递送(heart beat)”信息给群集中的服务器来避开此问题。服务器通过周期性经过网络或群集发送消息来如心跳般地递送信息包或消息。如心跳般地递送消息可包含的信息如，最近版本号，前一次版本号，或者实际的更新信息本身，如果更新信息足够小到可行的话。如果服务器接收到包含最近版本号的如心跳般地递送消息，而且服务器不在数据的该版本上或者没有接收到最近的使无效请求，服务器可从服务器请求或拉取使无效消息。

最初发送使无效请求的初始服务器，也可以是发送多点传送和/或如心跳般地递送的服务器，可为特定时间内存储最近请求，或者也可存储特定数量的最近请求。如果群集服务器请求初始服务器仍然存储着的使无效消息，初始服务器可使用诸如多点传送或点对点连接的方法来简单地发送消息到群集服务器。如果初始服务器不再有此消息，初始服务器可告诉群集服务器来简单地废弃它的整个缓存，其原因是它不可能告诉群集服务器哪个关键字发生了变化。群集服务器可从数据库读取新和/或当前的信息。这样会暂时地降低了性能，但新近缓冲的信息至少和数据库中的信息是一致的。

在实际操作中，客户或应用程序可通过读/写实体 bean 来更新数据项。此更新，或者包含多个更新的事务，会提交到数据库。使无效消息可和例如客户或服务器更新数据项所触发的消息一起被发送到群集中的服务器。不管是立即、随后或者当需要处理请求时，每个群集服务器都可以废弃本地缓存中的任何副本和从数据库读入新值。通常不可能读取未提交的数据，因此可能最好使用两步骤过程，首先提交数据，然后多点传送消息到群集。

上述办法的一个问题是它强迫客户来发起使无效请求，这会为客户引入

更多的复杂性。也有可能的是客户不正确地使用了使无效方法或者犯了错误。因此可能最好让系统自动来做。

如本发明所述的系统可通过使用“使无效目标”来处理此问题。使无效目标是基于如下想法，即只读和读/写 bean 指向数据库中相同的数据、而人们通过只读 bean 来读取数据并且人们通过读/写 bean 来更新数据。此想法是当读/写 bean 更新或修改时使只读 bean 无效。

当配置实体 bean 或企业级 Java Bean 时，通常有用来存储有关实际实体 bean 的元数据的配置描述符。例如，配置描述符可以是 XML 文档，用来提供关于 bean 所提供的、应可让客户访问的服务的信息。此信息可提供广泛类型的信息给客户，比如请求寻径信息，以及辅助性 Java 类的方法和类细节。可将标记加到配置描述符上，像先前那样作为“使无效目标”引用。读/写 bean 的使无效目标可包含任何关联只读 bean 的标识符。此使无效目标可用于当只读 bean 被更新时自动使任何关联的只读 bean 无效。

在一个实施例中，当服务器从读/写 bean 请求信息或产生只读 bean 来处理请求时，使无效目标可进行更新。当读/写 bean 从数据库或数据存储器转发信息到请求服务器时，读/写 bean 也可更新使无效目标。存储在服务器上包含读/写 bean 的 XML 文件可被更新为包含请求信息或创建 bean 的服务器的标识符。

图 3 再次展示了图 1 的系统 100，除了在此实施例中该系统还展示出利用使无效目标。当读/写 bean128 用来更新数据项 122 时，系统可查看与该读/写 bean128 相关联的使无效目标并发送使无效请求到与该读/写 bean 相关联的每个只读 bean130。使无效请求可直接由服务器 110 处理，其包含了读/写 bean128 和只读 bean130。使无效请求也可由任何适当的协议发送到包含在使无效目标中的只读 bean130 的其他服务器。一种方法是，服务器 110 通过点对点连接 134 直接联络服务器 112，告诉服务器 112 上的只读 bean130 来废弃服务器 112 上的缓冲副本 118。另一种方法是，服务器 110 可通过网络 106 发送多点传送消息 132 到任何服务器，该服务器包含在使无效目标中的只读 bean130，比如服务器 112。

对使无效后的只读 bean 的后续调用可导致比如企业级 Java BeanLoad 的方法被调用，它可从数据库读取当前信息到缓存中。例如，容器管理持久性 (CMP) bean，状态自动和数据库同步的实体 bean，可使用比如企业级 Java

Bean-jar.xml 的 XML 文件中的使无效目标元素来指定当读/写 bean 被修改时应使无效的只读实体 bean。在这个例子中的容器可自动将使无效目标使无效, 比如在事务完成之后。

这样的方法 200 如图 4 的流程图所示。数据项的副本存储在网络群集中的至少一个服务器上 202。标识信息, 或使无效目标, 提供给群集中服务器上的读/写 bean, 群集中任何服务器相关的信息包含只读 bean 和数据项的本地副本 204。数据项可使用读/写 bean 在数据库中更新 206。使无效请求可从包含读/写 bean 的服务器被发送到由标识信息所标识的任何服务器 208。然后, 接收到请求的每个服务器可废弃数据项的副本 210。

在这种方式中, 顾客或客户不需要编写任何额外代码来使项目无效。如本发明的实施例所述, 只有使无效目标必须被指定以便保持读/写和只读 bean 一致。Bean 可共存在相同的服务器上, 只读 bean 从本地缓存读取项目, 读/写 bean 从数据库读取或写入数据库。

为了改善性能, 如本发明所述的系统可一直等待到整个事务或系列更新都提交到数据库或数据表, 而不是为每个更新发送单独的消息。服务器, 比如发起更新的服务器, 可保持跟踪在事务期间哪些关键字被更新并且多点发送包括所有已更新主关键字的信息的单个消息。这样的消息批处理可改善这种系统的总体性能和可降低错误或不一致的机率。

如本发明所述的可用系统的一个例子包含股票符号的表, 以及每个符号关联的信息, 比如价格和数量。Java 服务端页面(JSP)可用来允许用户请求股票的当前价格。JSP 可从只读实体企业级 Java Bean 读取信息。Java 消息服务(JMS)队列可接收带有股票价格更新的消息。消息驱动 bean 可从队列中读出这些消息和更新关联的 CMP 实体 bean。当这个修改发生时, 容器可使关联的只读 bean 无效。

本发明的优选实施例的上述描述是为说明和描述用途而提供的。它不会很详尽或将本发明局限在所公开的精确形式里。对于本领域的技术人员来说, 明显地存在许多修改和变化。选择和描述的实施例只是为了更好地解释本发明的原理和它的实践应用, 从而使本领域的其他技术人员可理解本发明的各种实施例和适用于所计划的特定用途的各种修改。本发明的范围应由所附的权利要求和它们的等价物来定义。

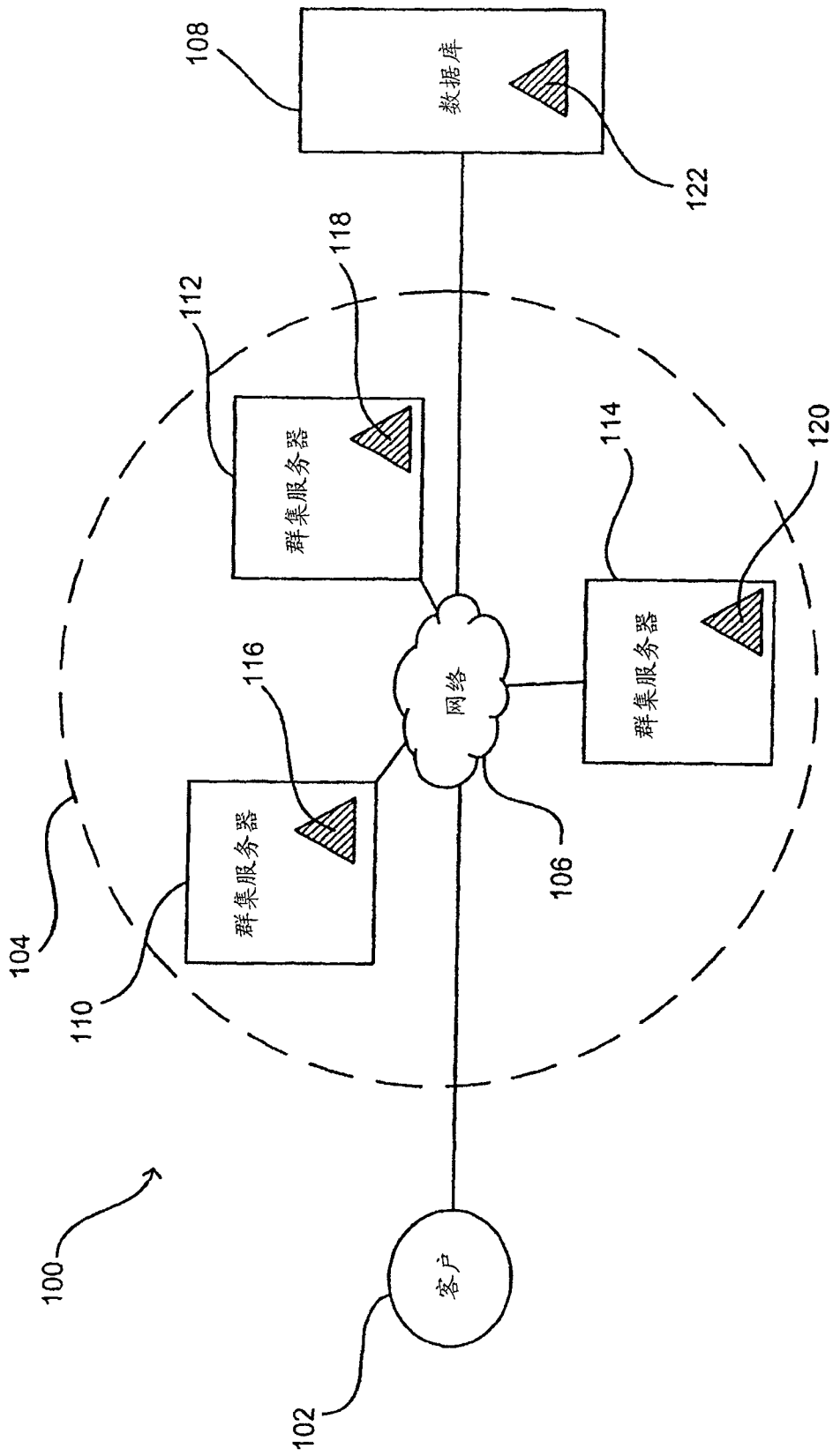
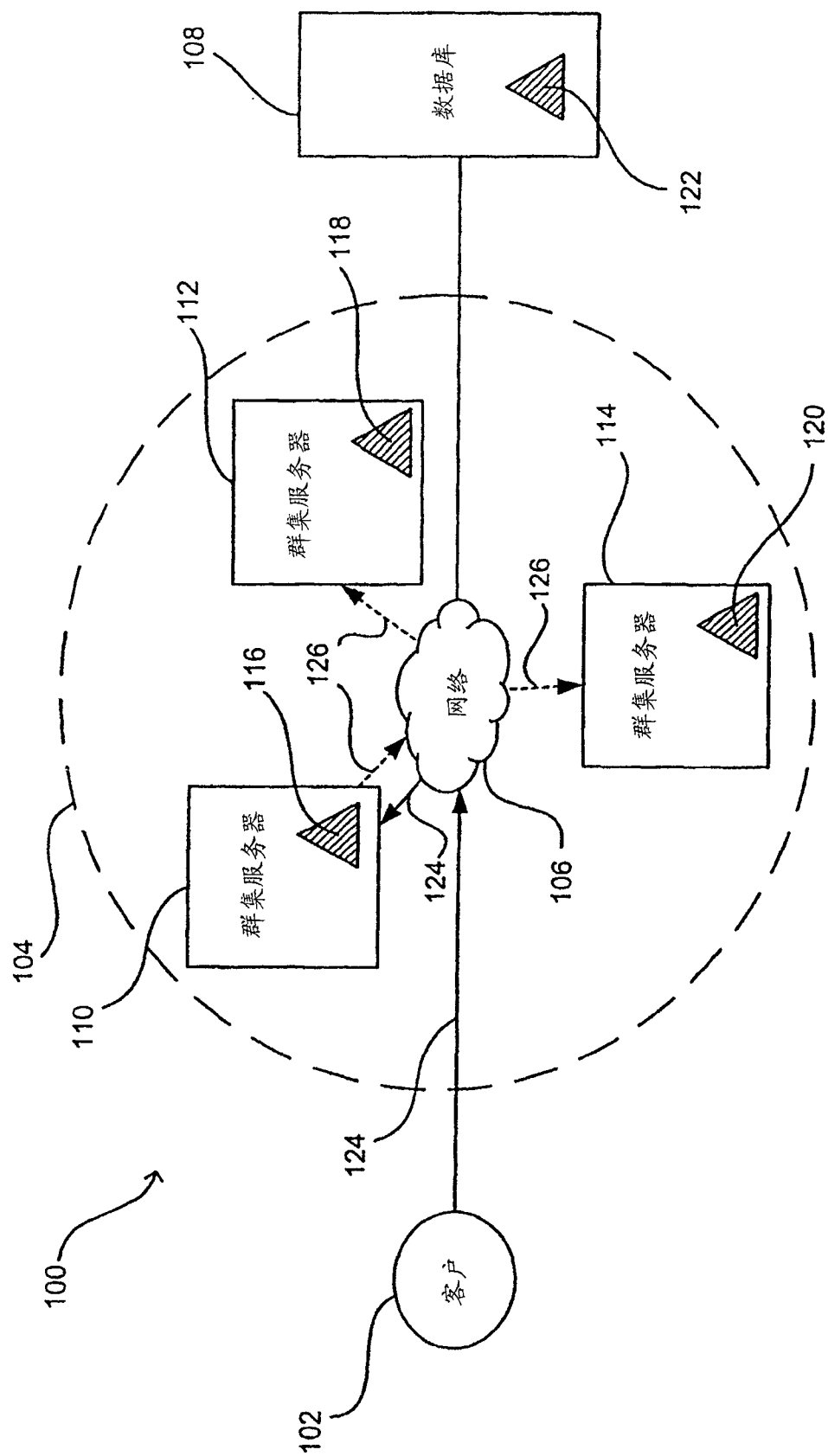


图 1



2
圖

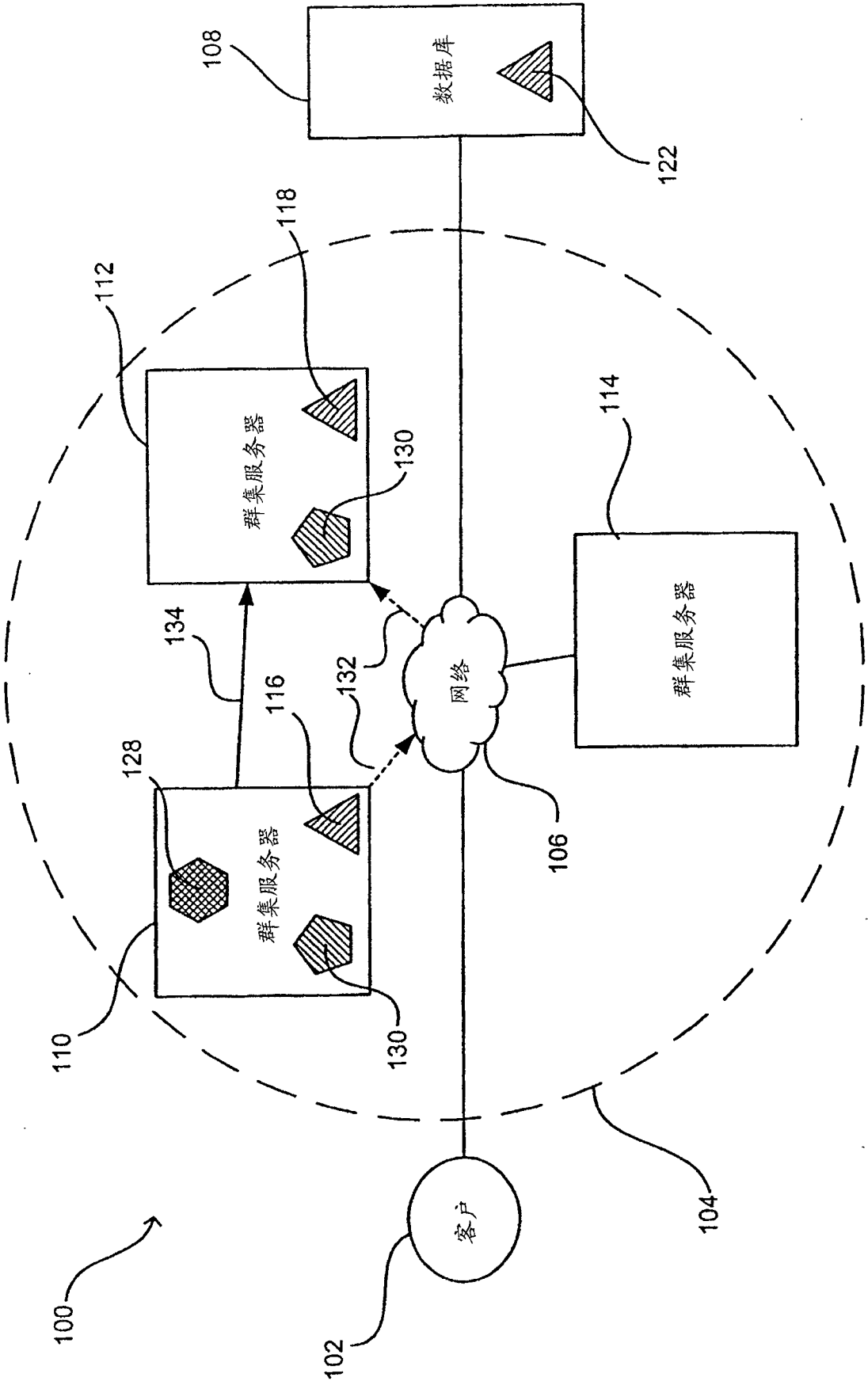


图 3

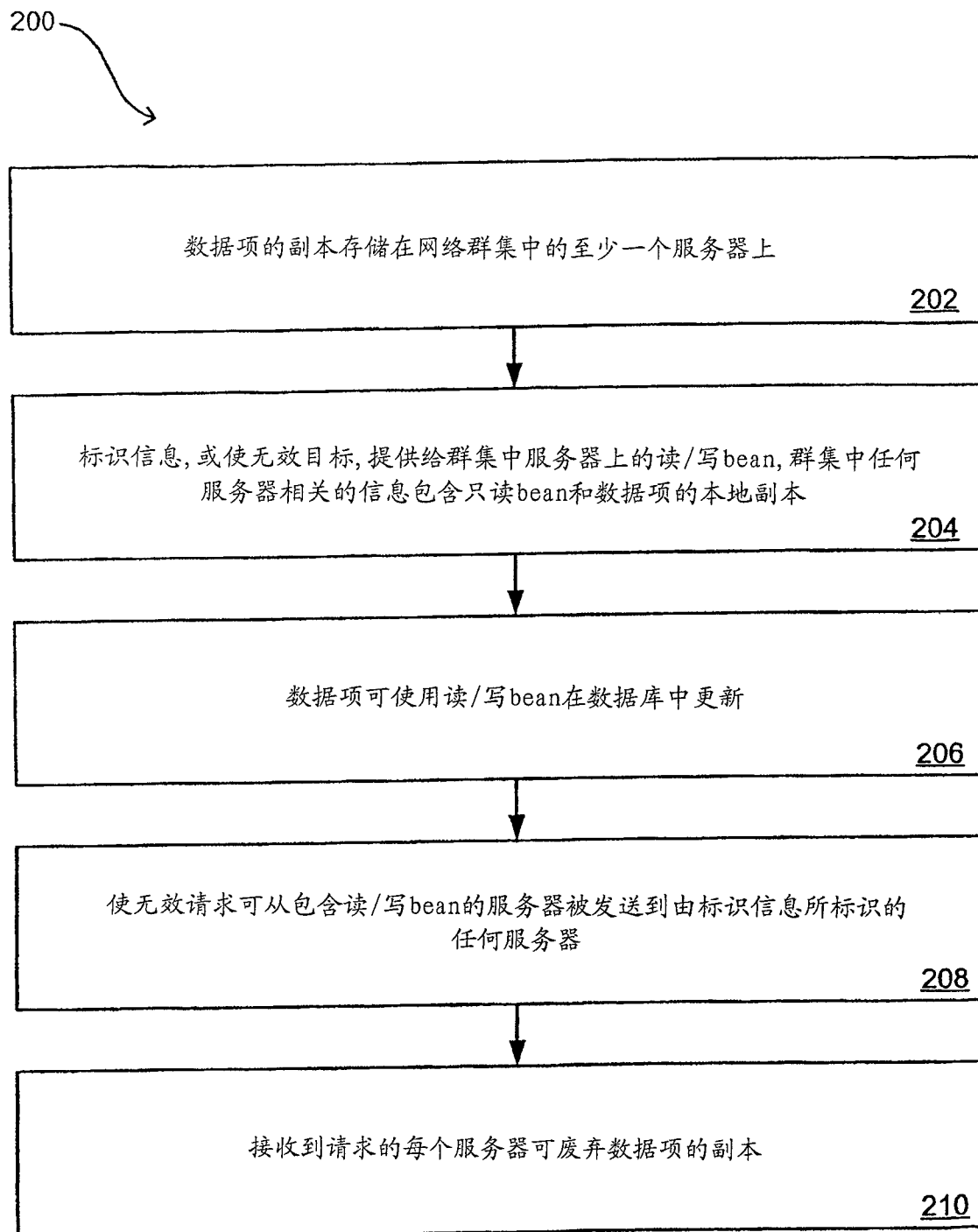


图 4

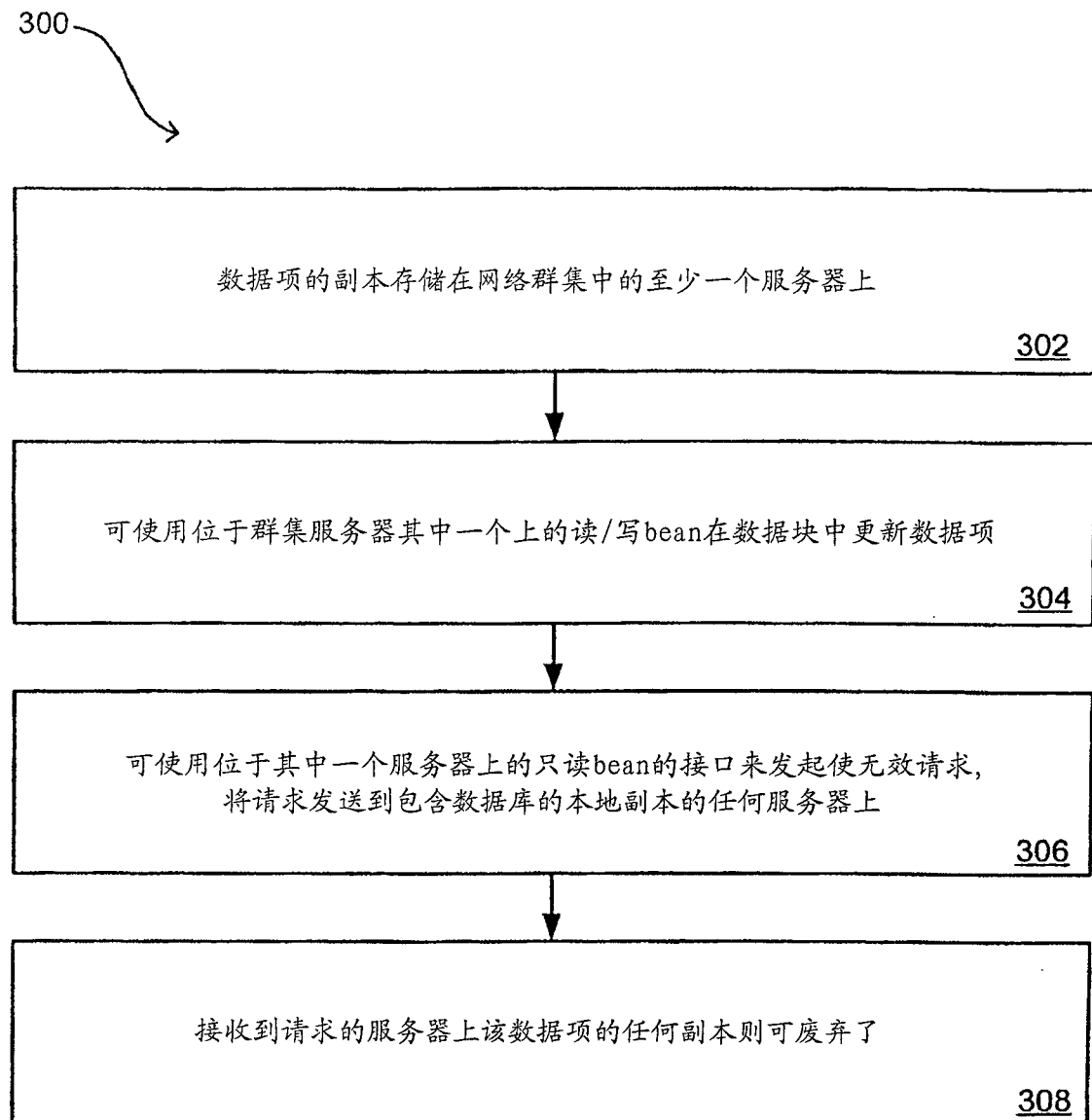


图 5

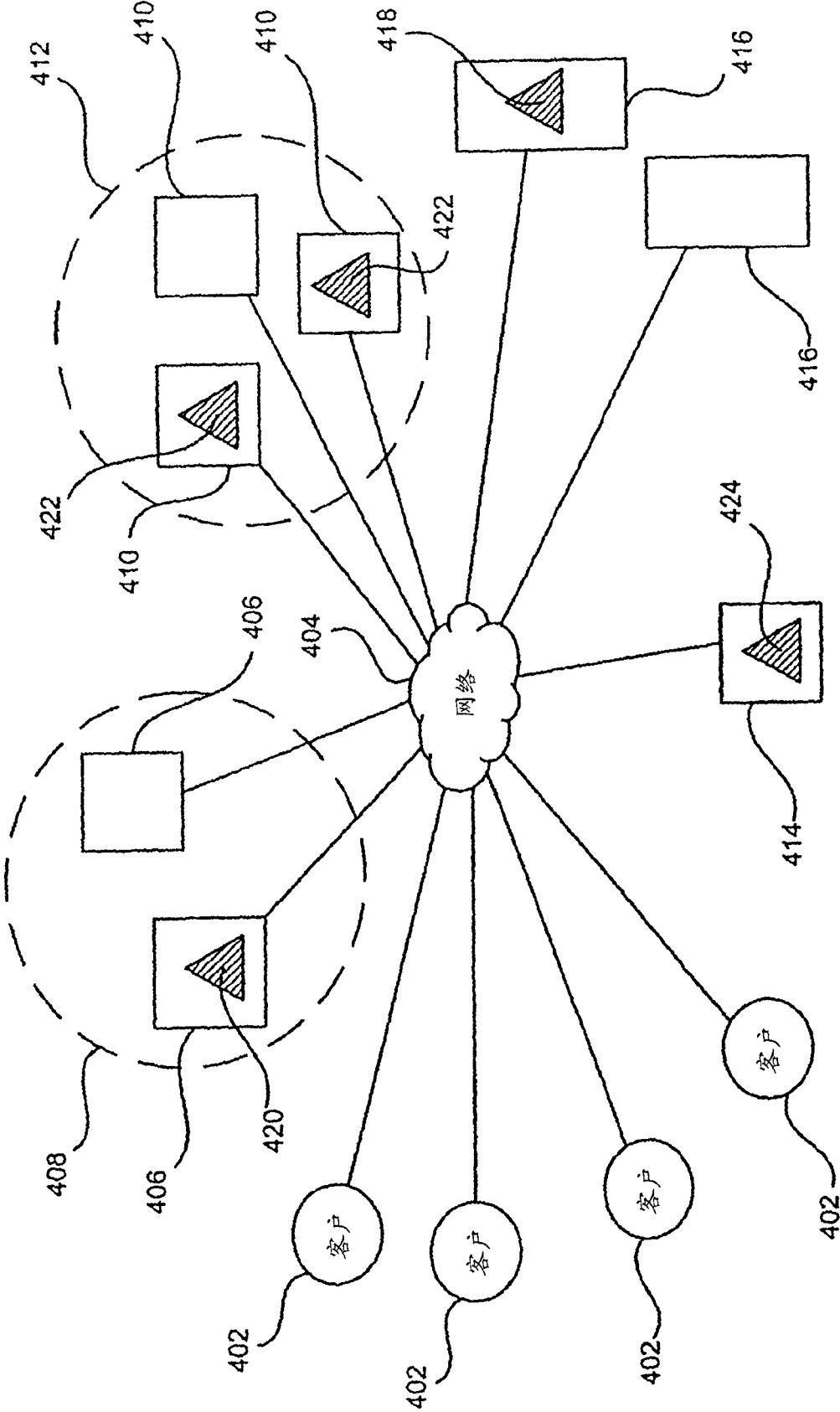


图 6