[19] 中华人民共和国国家知识产权局

[51] Int. Cl[7]
H04L 12/24

[12] 发明专利说明书

[21] ZL 专利号 98107795.1

[45] 授权公告日 2003 年 7 月 2 日

[11] 授权公告号 CN 1113503C

[22] 申请日 1998.4.29 [21] 申请号 98107795.1

[30] 优先权

[32] 1997. 5.30 [33] US [31] 866461

[71] 专利权人 国际商业机器公司

地址 美国纽约州

[72] 发明人 俞士纶

审查员 焦景梅

[74] 专利代理机构 中国专利代理(香港)有限公司

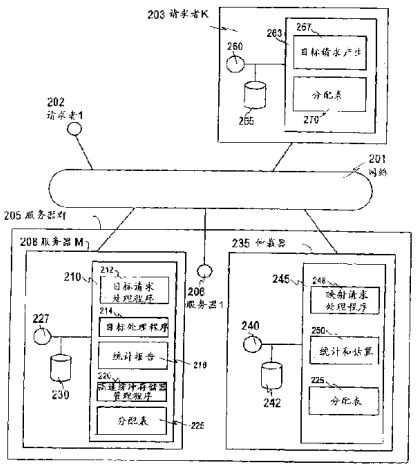
代理人 王 勇 王忠忠

权利要求书 5 页 说明书 16 页 附图 15 页

[54] 发明名称 因特网上的动态路由方法

[57] 摘要

在一个服务器集合或服务器群中目标请求的动态路由影响服务器的高速缓存效率和负荷平衡或仅仅影响负荷平衡。为提高服务器的高速缓存命中率，服务器选择影响请求目标的标识符(例如URL)。一种划分方法可以将目标标识符映射到类；并且请求者节点维护一个服务器分配表以便将每一个类映射到一个服务器选择。该类到服务器分配表可以随着工作负荷变化而动态改变并且还影响服务器能力。



ISSN1008-4274

1. 一种用于服务对目标的请求的一组服务器节点的动态路由方法，包括步骤:

用一个所请求的目标捎带元信息; 并且

根据该元信息动态修改关于一个服务器分配的路由信息。

2. 权利要求 1 的方法，其特征在于，还包括平衡服务器节点间负荷的步骤。

3. 权利要求 2 的方法，其特征在于，平衡服务器节点间的负荷的步骤还包括:

优化对所请求的目标的高速缓冲存储器命中。

4. 权利要求 1 的方法，其特征在于，该服务器集合包括因特网环境中的一代理服务器群或一 Web 服务器群。

5. 权利要求 1 的方法，其特征在于，还包括根据一个目标标识符的分级映射分配服务器的步骤。

6. 权利要求 5 的方法，其特征在于，该目标标识符是一个 URL。

7. 权利要求 3 的方法，其特征在于，还包括步骤:

将一个目标标识符映射为一个类; 和

基于该类和一个类到服务器分配表分配一个服务器。

8. 权利要求 7 的方法，其特征在于，所述的映射步骤还包括通过一个散列表将该目标标识符映射为类或散列类的步骤。

9. 权利要求 3 的方法，其特征在于，提供了在该请求者处一个服务器选择方法以便减少分配请求通信量，包括步骤:

在请求者节点维护一个类到服务器分配表用于服务器选择，包括步骤:

将每一个目标请求的目标标识符映射为一个类;

如果在该类到服务器分配表中得不到合理的服务器分配，那么向仲裁器发出一个映射请求; 和

为响应所述的映射步骤，修改该类到服务器分配表。

10. 权利要求 3 的方法，其特征在于，在该请求者处的以便减少分配请求通信量的一个服务器选择方法，包括步骤:

在请求者节点维护一个类到服务器分配表用于服务器选择，包括步骤:

将每一个目标请求的目标标识符映射为一个类；

如果在该类到服务器分配表中得不到合理的服务器分配，那么选择一个服务器；和

修改具有一个选定的服务器的类到服务器分配表，以响应所述的选择步骤。

11. 权利要求 3 的方法，其特征在于，所述的平衡该负荷的步骤还包括作为有关每个类的负荷的一个函数将类分配给服务器节点的步骤。

12. 权利要求 3 的方法，其特征在于，所述的平衡该负荷的步骤还包括将类递增地再分配给服务器的步骤。

13. 权利要求 1 的方法，其特征在于，所述的分段步骤包括使用一个 PICS 协议修改该路由信息。

14. 权利要求 1 的方法，其特征在于，还包括每一个请求节点传送一个关于一个目标的一个当前服务器分配的请求的步骤。

15. 权利要求 14 的方法，其特征在于，所述的发送一个请求的步骤基于该所请求的目标的类使用一个 PICS 协议确定当前服务器分配。

16. 权利要求 5 的方法，其特征在于，根据一个目标标识符的分级映射分配服务器的步骤，还包括步骤：

将每个群分配到一个虚拟的服务器节点；和

将该虚拟的服务器节点动态映射为一个实际的服务器节点。

17. 权利要求 16 的方法，其特征在于，该服务器节点集合包括一个域名服务器 （DNS ），其中所述的动态映射步骤包括一个名称到地址映射和与该名称到地址映射有关的一个活动时间周期 （TTL ），还包括步骤：

该 DNS 在小于 TTL 的间隔内将该虚拟服务器节点动态映射到该实际服务器节点；

将修改的服务器映射发送给所有服务器；

其中所述的元信息包括该修改的服务器映射；和

其中动态修改路由请求的所述步骤包括根据该修改的服务器映射发送后续目标请求的步骤。

18. 权利要求 16 的方法，其特征在于，该服务器节点集合包括一个 TCP 路由器，还包括该路由器将该虚拟服务器节点动态映射到一个

实际代理节点的步骤。

19. 权利要求 1 的方法，其特征在于，还包括一个异质请求者环境，其中并不是所有的请求者都适合于执行所述的动态映射步骤。

20. 权利要求 2 的方法，其特征在于，还包括根据一个目标标识符或 IP 地址的分级映射分配服务器的步骤。

21. 权利要求 2 的方法，其特征在于，还包括将一个请求者标识符映射到一个类和根据该类分配服务器的步骤。

22. 权利要求 2 的方法，其特征在于，还包括步骤:

该服务器向该请求者传送修改的元信息，和

该请求者修改该分配。

23. 一种在用于服务对目标的请求的多个代理服务器节点间的动态路由方法，包括步骤:

根据一个所请求的目标的目标标识符分配一个服务器;

将一个修改的服务器分配传送给一个目标请求者，以响应所述的分配步骤。

24. 权利要求 23 的方法，其特征在于，分配服务器的所述步骤包括根据该目标标识符的分级映射分配该服务器的步骤。

25. 权利要求 23 的方法，其特征在于，还包括步骤:

将一个目标标识符映射到一个类; 和

根据该类和一个类到服务器分配表分配一个服务器。

26. 权利要求 23 的方法，其特征在于，提供在该请求者处的服务器选择方法以便减少分配请求通信量，包括步骤:

在请求者节点处维护一个类到服务器分配表用于该服务器选择，包括步骤:

将每一个目标请求的目标标识符映射到一个类;

如果在该类到服务器分配表中得不到合理的服务器分配，则接着向仲裁器发出一个映射请求;

为响应所述的映射步骤，修改该类到服务器分配表。

27. 权利要求 23 的方法，其特征在于，提供在该请求者处的服务器选择方法以便减少分配请求通信量，包括步骤:

在请求者节点处保持一个类到服务器分配表用于该服务器选择，包括步骤:

将每一个目标请求的目标标识符映射到一个类；

如果在该类到服务器分配表中得不到合理的服务器分配，则接着选择一个服务器；

修改具有一个所选定的服务器的类到服务器分配表，以响应所述的选择步骤。

28. 一种在用于服务对目标的请求的多个Web服务器节点间的动态路由方法，包括步骤：

根据一个所请求的目标的标识符将指向同一主机名称或地址的目标请求分配给Web服务器群中的不同服务器；

将一个所修改的服务器分配传送给一个目标请求者；和

该目标请求者动态维护所请求的目标的该修改的服务器分配，用于后续的目标请求。

29. 权利要求28的方法，其特征在于，分配目标请求的所述步骤包括根据该目标标识符的一个分级映射分配该目标请求的步骤。

30. 权利要求28的方法，其特征在于，还包括步骤：

将一个目标标识符映射到一个类；和

根据该类和一个类到服务器分配表分配服务器。

31. 权利要求28的方法，其特征在于，提供在该请求者处的服务器选择方法以便减少分配请求通信量，包括步骤：

在一个请求者节点处维护一个类到服务器分配表用于服务器选择，包括步骤：

将每一个目标请求的目标标识符映射到一个类；

如果在该类到服务器分配表中得不到合理的服务器分配，则接着向仲裁器发出一个映射请求；

为响应所述的映射步骤，修改该类到服务器分配表。

32. 权利要求28的方法，其特征在于，提供在该请求者处的服务器选择方法用于减少分配请求通信量，包括步骤：

在一个请求者节点处维护一个类到服务器分配表用于服务器选择，包括步骤：

将每一个目标请求的目标标识符映射到一个类；

如果在该类到服务器分配表中得不到合理的服务器分配，则接着选择一个服务器；

修改具有一个所选定的服务器的类到服务器分配表，以响应所述的选择步骤。

33. 一种用于一个服务器节点集合的动态路由方法，其中可以将对该服务器节点集合的请求分配给该群中的不同的服务器，所述的方法包括步骤：

一个请求者定期向一个服务器发送映射请求，该映射请求包括一个请求者标识符或 IP 地址之一；

基于请求者负荷和服务器能力之一将所述请求者标识符或 IP 地址之一映射到服务器节点集合中的一个服务器；

向所有服务器发送一个服务器映射，以响应所述的映射步骤；和

如果一个服务器从不再分配给那个服务器的一个请求者接收了一个请求，则该服务器将请求者到服务器分配的改变通知该请求者。

34. 权利要求 33 的方法，其特征在于，所述通知请求者的步骤，还包括步骤：该服务器服务该请求。

35. 权利要求 33 的方法，其特征在于，还包括步骤：

将该请求者标识符 IP 地址划分成类；和

在仲裁者服务器和在服务器集合中维护一个类到服务器分配表。

36. 权利要求 35 的方法，其特征在于，该仲裁者服务器包括因特网环境中的一个 DNS。

因特网上的动态路由方法

本申请涉及共同未决的P.Yu申请的申请日为2/7/97、标题为“一种基于动态间隔平衡的方法和装置”的美国专利申请（序列号为08/798,385），和Dias等人申请的申请日（临时）为12/5/96、标题为“一种用于具有可选控制的负荷平衡的计算机系统和方法”的美国专利申请（临时）（序列号为60/031，849）。这些共同未决的申请和本发明一同转让给纽约Armonk的国际商业机器公司。在此结合这些共同未决申请中陈述的描述作为整体援引到本申请中。

本发明总的来说涉及提供在一个服务器集合（或群）（例如因特网中的代理服务器和环球网（Web）服务器）中的负荷平衡。本发明的一个更具体的方面涉及使用对客户请求的响应所分段的元数据修改路由信息的方法。而另一方面还涉及优化高速缓冲存储效率的一种负荷平衡方法。

尽管字典的含义也蕴涵了这里使用的某些术语，但是如下一些术语的汇编仍可能是有用的。

因特网

使用TCP/IP协议组的网络和网关的网络。

客户机

一个客户机是向服务器发命令的一台计算机，服务器执行与该命令有关的任务。

服务器

在收到另一台计算机的命令后执行一个任务的任何计算机都是服务器。一个Web服务器一般支持一个或多个客户机。

环球网（WWW或WEB）

通过撳按感兴趣的高高亮的词或短语让人们查询因特网上信息的因特网应用从服务器切换到服务器和从数据库切换到数据库（超链接）。一个因特网WWW服务器支持客户机并提供信息。可以把该Web看作具有如URL所寻址的所有资源的因特网，并且该网使用HTML显示与URL有关的信息而且向其它的URL提供了一个指向-并-撳按的界间。

全球资源定位器（Universal Resource Locator)(URL)

唯一识别或寻址因特网上的信息的一种方法可以认为是一个电子邮件地址或一个完全合格的网络文件名的一种 **Web** 文件形式。可以通过一个超级链接访问它们。一个 **URL** 的例子是“ **http://www.philipyu.com:80/table.html**”。这里该 **URL** 具有四个部分。从左边开始，第一部分指明所使用的协议，通过一个“ : ”与该定位器的其它部分分开。下一部分是主机名或目标主机的 **IP** 地址；它在左边通过“ // ”分界和在右边通过一个“ / ”或可选的一个“ : ”分界。端口号是可选的，并且在左边与主机名通过一个“ : ”分开，而在右边通过一个“ / ”分开。第四部分是实际的文件名称或程序名称。在该例子中，“ **.html** ”扩展意味着这是一个 **HTML** 文件。

超文本标记语言（ **Hypertext Markup Language** ）（ **HTML** ）

HTML 是由网服务器建立和连接 **Web** 用户可游览的文件所使用的语言之一。**HTML** 使用超文本文件。

超文本传送协议（ **HTTP** ）

HTTP 是无国籍协议的一个例子，意味着独立处理从一个用户到一个服务器的每一个请求。该服务器没有以前连接的记录。在一个 **URL** 的开始，“ **http:** ”表示应该使用 **http** 检索该文件。

因特网浏览器或 **Web** 浏览器

运行因特网协议如 **http** 并且在用户屏幕上显示结果的一个图形界间工具。该浏览器可以作为一个因特网漫游向导，以图形桌面，目录以及当用户在因特网上“ 漫游（ **surfs** ） ”时所使用的查找工具进行。在该申请中 **Web** 浏览器是与环球网通信的一种客户机服务。

客户机高速缓冲存储器

客户机高速缓冲存储器一般用作被客户机访问的目标的主高速缓冲存储器。在一个 **WWW** 环境中，客户机高速缓冲存储器一般由 **Web** 浏览器提供并且可以高速缓存在当前调用期间访问的目标，即一个非持久性的高速缓冲存储器，或者可以高速缓冲存储经过调用的目标。

高速缓冲存储器代理

网络中的专用服务器作为代表客户机定位一个目标的高速缓冲存储的拷贝的代理。高速缓冲存储器代理一般作为二级或较高级高速缓冲存储器，因为，来自客户机的高速缓冲存储器的高速缓冲存储器-失中导致调用它们。

HTTP 守护程序(Daemon)(HTTPD)

具有超文本标记语言和公共网关接口能力的一个服务器。HTTPD一般由给内部网上的机器提供硬件连接和对因特网提供访问的访问代理支持，例如TCP/IP耦合。

环球网上的通信量正以指数级增长。代理服务器，特别到一个大的机构或地区的网关，可以包括一个计算节点集合。同样地，在一个受欢迎的 （热门的 ）Web站点，使用一个计算节点集合 （或群 ）以支持访问要求。

为获得一个服务器群中的优良性能，应该在节点集合中平衡负荷。应该通过集中相同目标请求，优化群中的一个给定服务器中的高速缓冲存储器命中率的要求来减轻负荷。

在一个多处理器或多节点环境如IBM S/390 Sysplex中关于负荷平衡的以前的工作主要集中于为每个叫入任务或用户会话选择多种通用资源之一的调度算法。调度程序控制每个叫入任务或会话的调度并且不对资源选择进行高速缓冲存储。

在地域上广泛分布的同样的站点中平衡负荷的一个公知方法叫作循环域名服务器 （RR-DNS ）方法。在1994《计算机网络和ISDN系统 》，卷27、第68-74页的Katz.,E.,Butler,M., 和McGrath,R.的标题为“一种可缩放的HTTP服务器:NCSA原型 ”论文中，该RR-DNS方法用于平衡在一组Web服务器中的节点。这里，分布的站点组由一个URL （例如www.hostsite.com ）表示；使用子域名服务器定义该分布地点的一群子域。子域名服务器以循环方式把名称解析请求映射为不同的IP地址 （在分布的群中 ）。因此，将把客户机的子集指定到每个重复的站点。为减少网络通信量，并不对每个服务请求发出一个映射请求。而是在一个 “活动时间 ” （TTL)间隔中存储该映射请求的结果。在TTL间隔内，发出的连续请求保持以前的映射并且因此将被发送到同一服务器节点。

RR-DNS方法带来的问题是可能产生在分布站点中的负荷不平衡（见例如 1996 IEEE国际计算机学会国际会议 （COMPCON ）会刊.第41期 Dias,D.M.,Kish,W.,Mukherjee,R.和Tewari,R.的 “一种可缩放的和高有效的Web服务器 ” ）。负荷不平衡可以由网络中的各种网关、防火墙和域名服务器的名称和IP地址之间关联的高速缓冲存储引

起。因此，对于 TTL 周期，通过这些网关、防火墙和域名服务器发送的所有新的客户请求将分配给存储在该高速缓冲存储器的一个站点。本领域的技术人员将意识到 TTL 值的简单减少将不能解决该问题。实际上，许多名称服务器经常不接受低的 TTL 值。更重要的是，TTL 值的简单减少不能减少由不规则分布的客户请求速率引起的负荷的偏斜 。

在一个本地节点群内负荷平衡的一种方法是使用正如在 1992 IBM 研究报告 RC 18442 中、Attanasio, Clement R. 和 Smith, Stephen E. 的“ 由一个松散耦合的计算机密闭群实现的虚拟多处理器 ”和在此结合作为整体援引的发表在 1994 年 12 月 6 日标题为“ 用于把一个计算机群看作一个单一主机的方法和装置 ”的美国专利（专利号为 5,371,852 ）中所描述的一种所谓的 TCP 路由器。这里，只给客户该 TCP 路由器的地址；该 TCP 路由器或者以循环方式，或者根据节点负荷在该群的节点中分配叫入请求。应该注意到这种 TCP 路由器方法限于一个本地节点群。

最近，在 1997 年 1 月 IBM 研究报告 RC 20680 中的在此结合作为整体援引的 Colajanni,M.,Yu,P 和 Dias,D. 的标题为“ 分布的 Web 服务器的调度算法 ”的文件中，描述了一种多层循环方法，把网关根据请求速率分成多层。利用一个循环算法分别处理来自每层的请求。该方法也可以处理一个同质的分布式服务器结构。

在上面的所有方法中，目的在于平衡一个服务器集合中的负荷。动态路由判定不考虑被请求目标的相同性。换句话说，对于同一个目标的多个请求可以发送给不同的服务器以便平衡负荷。这将导致差的高速缓冲存储器命中率，对于代理服务器特别严重，因为参照的不同 Web 页数可能是很大的。尽管在一个 Web 服务器群中，可以对 Web 页进行静态划分，其中为每个划分分配一个不同的（ 虚拟的 ）主机名称或 IP，静态划分的方法缺乏应付动态负荷变化的灵活性并且还是不可缩放的。

因此，需要在一个服务器群中改进的负荷平衡方法和装置，它不仅平衡该群中的负荷，而且利用集中相同目标请求来优化该群中一个给定的服务器的高速缓冲存储器命中率。本发明致力于解决这样一种需求。

还需要改进的路由方法，它根据工作负荷状况动态分配每个服务器处理一个目标空间子集并且将目标请求发送给分配给与该目标相关的

子空间的服务器。本发明还致力于解决这样一种需求。

根据前面提到的需求，本发明目的在于提供同时考虑：服务器的高速缓冲存储效率和负荷平衡；或仅考虑负荷平衡的用于在服务器集合中动态传送目标请求的一种改进的方法和装置。

本发明也具有能够通过用对路由请求的响应“捎带”(piggyback)元信息来动态修改服务器路由信息的特征。本发明具有能够根据所请求的目标的标识符（例如URL）映射服务器并且如果工作负荷情况改变则动态修改该映射来提高该服务器的高速缓冲存储器命中率的其它特征。在一个因特网环境中，服务器群可以包括，但是不限于一个代理服务器群或一个Web服务器群。

具有本发明的用于在一组服务器节点中动态发送目标请求的特征的方法包括步骤：用所请求的目标捎带元信息；并且根据元信息为服务器分配动态修改路由信息。

具有本发明的用于在一组服务器节点中在优化高速缓冲存储器命中率的同时动态发送目标请求的特征的方法，还包括步骤：将一个目标标识符映射到一个类；并且根据该类和类到服务器分配表分配服务器。

本发明还具有其它特征，可以根据类到服务器分配的动态改变在“提出要求时”能够通知请求者节点。该类到服务器分配可以随着工作负荷变化而改变。为避免花费很大将该改变对所有可能请求者广播，该服务器可以有利地继续服务目标请求，即使该服务器不是分配处理那个类的服务器。但是，该服务器可以在返回目标（或响应）的首部中表示新的类到服务器分配的信息。

另外，本发明的用于用所请求的目标捎带元信息的特征还可以应用于因特网的常规DNS路由以提高服务器群中的负荷平衡。这应该不同于使用目标的URL（或目标类）进行服务器分配（以便提高高速缓冲存储器命中率）的原理。DNS路由具有用于地址映射的一个合理的间隔（TTL）。本发明具有允许在小于TTL的间隔内产生服务器分配并且因此更好地反映真正的负荷状况的特征。服务器分配的改变可以用返回的目标捎带，避免增加通信量，以便将来的请求可以发送给新的服务器。

本发明还具有其它特征，能够根据工作负荷需求动态地和递增地改

变类到服务器分配以平衡负荷。

根据本发明的其它特征，在一个因特网环境中，PICS 协议可以用于传递各种类型的信息。当一个请求指向基于一个废弃的类到服务器映射项的一个服务器时，PICS 可以被服务器用来捎带关于一个新的类到服务器映射的元信息。PICS 也可以由请求者使用以便查询当前的类到服务器映射的协调者。

本领域的技术人员知道本发明可以应用于一般的分布式环境以及环球网。

本发明的这些和进一步的目的，优点和特征将会在下面的一个优选的实施例和附图的详细描述中更加清楚，其中：

图 1 是适用于本发明的一个因特网环境图；

图 2 是具有本发明特征的一个一般环境的更为详细的例子；

图 3 是 “类到服务器 ” 分配表的一个例子；

图 4 是图 2 的服务器逻辑的一个例子；

图 5 是服务器的目标处理程序的一个例子；

图 6 是服务器的目标处理程序的动态再分配程序的一个例子；

图 7 是服务器的目标请求处理程序的一个例子；

图 8 是服务器的统计报告程序的一个例子；

图 9 是图 2 的仲裁器逻辑的一个例子；

图 10 是仲裁器的统计和估算程序的一个例子；

图 11 是仲裁器的统计和估算程序的再分配程序一个例子；

图 12 是仲裁器的映射请求处理程序的一个例子；

图 13 是图 2 的请求者逻辑的一个例子；

图 14 是请求者逻辑的目标请求产生的一个例子；和

图 15 是包括一个域名服务器 （DNS ）的图 1 的服务器群的一个例子。

图 1 是适用于本发明的因特网环境图。如所描述的那样，可以包括任一一般的代理服务器节点 （118，125-127 ）的请求者 （110-153 ），客户工作站和个人计算机 （也叫做 PC ） （110，112，120，150-153 ）连接到网络 （105 ）。代理服务器，工作站和 PC 是本领域公知的。代理服务器节点的一个例子是由 IBM 出售的因特网连接服务器 （ICS ）。请求者通过网络 （105 ）请求来自服务器群

（103 ）的服务。该网络的例子包括，但不限于，因特网，环球网，内部网和局域网（LAN ）。服务器群包括多个服务器节点（161-163 ），以便处理高的通信需求。它既可以是一个代理服务器也可以是一个 Web 服务器群。该群中的服务可以包括，但不限于，诸如 IBM 以商标 S/390，SYSPLEX，SP2 或 RS6000 工作站出售的产品。因为一般地，每个请求可以由群中的任意服务器处理。典型的服务请求包括环球网页访问，远程文件传送，电子邮件和事务支持。

尽管在原则上请求可以由任意的处理器节点处理，发送关于相同目标的请求到一个服务器节点将导致在相同处理器节点较好的高速缓冲存储器命中的概率，和由此的优良性能。正如下面将描述的，本发明具有不仅在群中的处理器节点中平衡负荷，而且还能实现高的高速缓冲存储器命中概率的特征。

作为概述，根据本发明的路由方法在选择服务器处理该请求时，使用目标的一个逻辑标识符或符号名 （如 URL ）。另外提供一种划分（partition ）方法以便把目标标识符映射到类；并且请求者节点优选保持类到服务器分配表 （图 3 ）以便把每个类映射到服务器选择。一个优选划分方法使用一个常规的散列函数以便把一个目标 URL 散列到给定个数的散列类。优选由仲裁器 235 （图 2 ）将该散列函数赋予所有参加的服务器和请求者节点并且使该散列函数对所有参加的服务器和请求者节点是公知的。

仲裁器 235 监视每个服务器的负荷并且动态修改类到服务器分配以便提高负荷平衡。本发明还提供一种方法由服务器 103 在对类到服务器分配进行动态修改时通知提出要求的请求者节点。

一个来自请求者节点的请求在到达服务器群 103 前可能需要经过若干个中间请求者节点 （即代理服务器 ）。例如，节点 150 在到达服务器群 103 之前需要经过两级代理节点，125 和 118。如果该服务器群是一个代理服务器群，那么该服务器选择优选由与代理服务器群 103 最近的请求者 110-120 完成。在一个 Web 服务器的情况下，Web 服务器选择可以在路径上的中间请求者上完成。

本发明还具有使用 “捎带的 ” 元数据在请求者和服务器节点之间有效传递路由信息的特征。在一个 HTTP 实现中，通过使用现有的 Web 协议可以在目标首部中包括信息交换作为元-数据。PICS（ “因

特网内容选择平台 ” ）指定了发送有关电子内容的元信息的一种方法。PICS是一个Web联合协议建议（见http://www.w3.org/PICS)。PICS最初用于发送基于值的额定值标识符，例如“有多少披露部分与该内容有关，”但是该元信息的格式和含义完全是通用的。在PICS中，根据“额定值服务”或者该信息的生产者-和-企图的-用途对关于电子内容的元信息分组，并且在这样一个组里，可以发送任意多个类或维的信息。每个类具有一个允许值的范围，并且对于一条特定的内容，一个特殊的类可以具有一个单一值或多个值。另外，元信息组（称为“PICS标识符“ ）可以包括终止信息。还有允许一个PICS标识符适用于一条以上电子内容的机制。可以增加或从该内容中除去关于一条特定的电子内容的每一个PICS标识符。

例如，一个图象文件可以从一个具有单独的PICS标识符的服务器发送，该标识符的“额定值服务”字段根据“安全漫游”（SafeSurf）额定值系统表示它包括基于值的额定值标识符。根据本发明，当它经过一个企业代理时，该图象文件还可以接收第二PICS标识符，该标识符的“额定值服务”字段表示它包括类到服务器分配信息。当它经过一个部门代理时，可以除去第二PICS标识符。因此，客户计算机仅可以看见第一PICS标识符。HTTP协议已经增加了支持PICS的请求首部和响应首部。定义其它公共应用协议的技术团体，如NNTP，现在也正在考虑增加PICS支持。作为这些协议的一部分，所希望的PICS标识符类型表可以与一个请求包括在一起。PICS还指定了从一个中心标识符局服务器接收PICS信息的一种查询格式。一个PICS标识符样本是：（PICS-1.1 “http://the.rating.service”表示“http://the.content”代表“1997.07.01T08:15-0500” r(n 4 s 3 v 2 l 0)),其中‘n’‘s’‘v’‘l’是各种元信息类型的发送名称，并且该内容的适当值是4（对于n），3（对于s），2（对于v）和0（对于l）。只有识别标识符ID“http://the.rating.service”的软件知道如何解释这些分类和值。

在优选实施例中，使用两个不同类型的PICS标识符。第一种PICS标识符，称作“再分配”标识符或（R-label），由群中的服务器节点使用以便表示关于所返回的目标的目标类的“当前”服务器分配。第二种PICS标识符，称作“分配”标识符或（A-label），由请求

者使用以便确定来自在该情况下提供标识符局功能的仲裁器的目标的URL的“当前”服务器分配。

图2描述了具有本发明特征的网络（201）和系统的一个更为详细的例子。如所描述的，请求者节点（202-203）用于表示通过网络（201）能够发出请求的一个计算节点。该请求者节点优选包括CPU（260），存储器（263），例如RAM，和存储设备（265），诸如DASD或磁盘，和/或其它稳定的磁，电或光的存储器。存储器（263）根据本发明存储请求者203逻辑（参照图13描述的细节），优选实现为计算机可执行的代码，该代码可以从一个稳定的程序存储器（265）装入存储器（263）用于由CPU（260）执行。本领域的那些技术人员还将知道请求者（203）逻辑也可以通过网络（201）下装给请求者用于由CPU（260）执行。请求者203逻辑包括一个目标请求产生程序（267）（具有在图14描述的细节）并且维护该类到服务器分配表（270）的一份拷贝。

仲裁器（235）表示可以监视服务器业务量并且对“类到服务器”分配做出判断的任一普通的计算节点。仲裁器（235）优选包括CPU（240），存储器（245）例如RAM，和存储设备（242）例如DASD和/或其它稳定的磁，电或光的存储器。存储器（245）存储本发明的仲裁器逻辑（具有在图9描述的细节），优选实现为计算机可执行的代码，该代码可以从一个程序存储器（242）装入存储器（245）用于由CPU（240）执行。为清楚起见并且仅作为一个例子，把该仲裁器逻辑分成几部分，包括：一个映射请求处理程序（248），和一个统计和估算程序（250）。这些部分将参照图12和10分别详细描述。所维护的主要的数据结构是类到代理分配表（225）。类到代理分配表（225）的操作将与各个部分一起说明。

服务器1…M（206-208）可以包括任意常规的可以处理服务请求例如提供数据/目标访问和/或由请求者（203）请求文件传送的计算节点。服务器节点（208）包括CPU（227），存储器（210）和存储设备（230）例如DASD和/或其它稳定的磁，电或光的存储器。存储器（210）存储本发明的服务器逻辑（具有在图4描述的细节），优选实现为计算机可执行的代码，该代码可以从一个存储器（230）装入存储器（210）用于由CPU（227）执行。为清楚

起见并且仅作为一个例子，把该服务器节点逻辑分成若干部分，包括：一个目标请求处理程序 （212 ），一个目标处理程序 （214 ）和一个统计报告程序 （218 ）。这些部分将参照图 7，5 和 8 分别详细描述。它还包括一个高速缓冲存储器管理程序 （220 ）并且维护类到服务器分配表 （225 ）的一份拷贝。

图 3 提供了分配表 （225，270 ）关于 N=16 和 M=3 的一个例子，其中 N 优选是目标类的个数，即散列表或分配表的大小，并且 M 是服务器的个数。令 C（.）是分配表 （225，270 ），它为服务器 C（k）分配类 k。再参照图 2，不仅仲裁器 （235 ）和群中的每一个服务器 （206，208 ）节点，而且请求者节点 （202，203 ）都能够维护分配表 （225，270 ）的一份拷贝。在请求者处的表 （270 ）一般不是最新的，即不与服务器 （208 ）或仲裁器 （235 ）分配表 （225 ）同步。本发明具有不需发送昂贵的修改信息以便保持表同步的特征，并且优选使用捎带的元数据修改 “所需要的 ” 类到服务器映射。

图 4 描述根据本发明在 CPU（227 ）中执行的存储在存储器 （210 ）中的服务器 （208 ）逻辑一个例子。令 C （. ）是分配表 （225，270 ），它把类 k 分配给服务器 C （k ）。如在步骤 410 所描述的，服务器等待输入。在步骤 415，根据接收的输入，将采取不同的操作。如果接收的输入是一个目标请求，则在步骤 420 调用目标请求处理程序 212。将参照图 7 描述目标请求处理程序的一个更为详细的例子。在步骤 430，如果接收的输入是一个目标，则在步骤 440 调用目标处理程序 214。将参照图 5 描述目标处理程序的一个更为详细的例子。在步骤 445，如果接收的输入是一个统计收集请求 （来自仲裁器 ），则在步骤 460 调用统计报告程序(218)。将参照图 8 描述统计报告程序的一个更为详细的例子。在步骤 450，如果接收的输入是一个分配表更新请求 （来自仲裁器 ），则将在步骤 465 相应更新 C （k ），k=1，…，M。对于不是本发明焦点的其它类型输入 （例如一般的 HTTP “拉 ” 请求或一个 FTP 请求，可以调用一个适当的杂项处理程序 （470 ）。

图 5 描述目标处理程序 （214 ）的一个例子。在步骤 510 如果接收的目标类 （见图 7 步骤 750 ）属于如分配表表示的分配给该服务器的一个类，则在步骤 515 调用高速缓冲存储器管理程序 220。高速缓冲存储器管理程序确定是否应该高速缓冲存储该目标，并且如果是，则替

换当前高速缓冲存储的目标。然后，在步骤530，将该目标返回给请求者。如果在步骤510，所接收的目标的目标类不属于分配给该服务器的类（如该分配表所表示的），则接着在步骤520调用动态再分配程序。将参照图6描述动态再分配程序的一个更为详细的例子。

图6描述动态再分配程序（步骤520）的一个例子。如步骤610所述，从分配表（图3）确定用于处理该目标类的适当的服务器标识符id（或IP地址）。在步骤620，把一个R-标识符插入到该目标的首部，其中目标的分类值表示分配给处理该目标类（图3）的服务器。

图7描述目标请求处理程序212的一个例子。如步骤710所述，如果在本地缓存器中发现该目标，则在步骤720检查分配表（225）以便确定该目标类（图3）是否由这个处理器（图3）处理。如果不是，则调用（图6）动态再分配程序。在步骤740，将该目标返回给请求者。在步骤710，如果发现该目标还没有在本地高速缓冲存储，则在步骤750发送一个请求以便得到该目标（代表请求者）。

在下面的描述中，令CS（j,i）是对由服务器j接收的类i中的目标的请求个数（在当前的测量间隔内）；并且令CA（i）是对由所有服务器接收的类i中的目标的请求总数。而且，指定SA（j）作为分配给服务器j的目标类的请求总数。

图8描述统计报告程序（218）的一个例子。如步骤810所述，服务器j发送它的负荷信息CS（j,i）给仲裁器(i=1，…N)。在步骤820中，将CS（j，i）,(i=1，…，N)复位为零，即启动新的收集或测量间隔的记数。

图9描述仲裁器逻辑（235）的一个例子。在步骤910，仲裁器等待输入。在步骤920，如果检测到一个映射请求，则调用映射请求处理程序（248），在步骤940（将参照图12描述映射请求处理程序（248）的一个详细例子）。在步骤930，如果检测到统计收集间隔的计时器到时，则仲裁器在步骤950（将参照图10描述统计和估算程序（250）的一个详细例子）执行统计和估算程序（250）。在步骤960，利用所更新的分配表将更新请求传送给所有的服务器。

图10描述统计和估算程序（250）的一个例子。如步骤1010所述，统计收集请求从服务器j，(j=1,…,M)传送到所有服务器以便得到CS（j,i）(i=1,…,N)。在步骤1020，为每一个类计算CA(i)（关于

每个类 i 在所有服务器上的请求的总数 ）。在步骤 1030，为每一个服务器 j 计算 CA (j)（ 分配给每个服务器 j 的类的请求的总数 ）。在步骤 1040，计算服务器负荷的上阈 TH。TH 优选定义为平均负荷上的一个百分率（ d ）。例如， 可以是 0.2，意味着，负荷平衡的目的是没有服务器的负荷超过平均的 20 %。在步骤 1050，如果任一服务器的负荷超过上阈 TH，则调用再分配程序以便调节类到服务器分配使得能够实现较好的负荷平衡。将参照图 11 描述再分配程序的一个详细例子。在步骤 1070，将统计收集计时器复位为所需的统计收集间隔的长度。

图 11 描述再分配程序（ 步骤 1060 ）的一个例子。在步骤 1110，TO 包括超过负荷上阈（ TH ）的服务器集合。在步骤 1115，令 k 是 TO 中最大负荷服务器的索引。在步骤 1120， TU 包括没有超过负荷阈值的服务器集合。在步骤 1125，令 l 是 TU 中最少负荷服务器的索引；并且在步骤 1130，令 i 是分配给具有最小类负荷 CA（ i ）的服务器 k 的类。在步骤 1135，如果对服务器 l 的再分配类 i 没有引起服务器 l 的负荷超过阈值，即 SA（ l ）+CA（ i ）〈 =TH，则在步骤 1140（ 通过将 C(i) 改变为 l 并且改变 SA(l)和 SA(k)), 将类 i 从服务器 k 再分配给服务器 l；并且在步骤 1145，更新 SA（ l ）和 SA（ k ）以便反映该类的再分配。具体地说， SA（ l ）增加 CA（ i ）而 SA（ k ）减少 CA（ i ）。否则，在步骤 1160，从 TU 中删除服务器 l，因为它不再能够接收来自过载服务器的负荷。在步骤 1150，如果服务器 k 的负荷仍然超过阈值，即 SA（ k ）〉 TH，则再次执行步骤 1130。否则，在步骤 1155，从 TO 中删除服务器 k，因为它的负荷不再超过阈值。在步骤 1170，如果 TO 不空，则再次执行步骤 1115。在步骤 1165，如果 TU 不空，则再次执行步骤 1125。

本领域的那些技术人员将容易理解在本发明的精神和范围内可以使用各种可替换的实施例和种种对本发明的扩展。例如，在步骤 1140，再分配是一个简单的贪婪方法以便允许一个单一类（ 图 3 ）从服务器 k 移动到服务器 l 以减少负荷的不平衡。如果它能够提高负荷平衡，则允许关于从服务器 k 的一个类与服务器 l 的另一个类调换或交换的扩展。在步骤 1135，只有当服务器 l 不超过负荷阈值时才进行再分配。可以放松该标准以便代之以测量总的过载是否减少。而且，如果任一类负荷 CA（ i ）超过 TH，则可以将它分配给多个服务器，其中这些服

务器的每一个将得到关于那个类的请求的百分率。仲裁器可以根据分配给该服务器的百分率随机地将服务器分配给关于那个类的请求者。在服务器（ 208 ）可以实现类似的再分配。

而且，在优选实施例中，假定群中的所有服务器具有相同的处理能力。本领域的那些技术人员将容易理解它可以容易地扩展为包括异质的服务器。在异质服务器的情况下，可以归一化负荷平衡，以便反映由处理能力划分的接收的请求数。具体地说，可以通过服务器 j 的处理能力归一化 SA（ j ）。

注意到图 11 描述了对类到服务器分配进行的动态增加改进方法的一个例子 。本领域的那些技术人员将会认识到在本发明的精神和范围内可以使用许多可选的提供初始的类到服务器分配表的方法。如果得不到以前的工作负荷信息 ，则可以使用一种随机的或循环的类到服务器分配。否则，可以使用最少处理时间优先（ LPT ）算法。以类的访问负荷的降序对类分类。从该表中首先除去该表顶部的类（ 即具有最重负荷的类 ）并且将它分配给目前分配的最少负荷的服务器。然后相应调整那个服务器的分配负荷。重复该处理直到分配了所有的类。

图 12 描述了映射请求处理程序（ 245 ）的一个例子。如步骤 1210 所述，将目标 id(例如 URL)映射为它的类，例如通过常规的散列或其它方法。例如，这可以通过对 URL 逻辑的前 4 个字节与该 URL 的后四个字节的逻辑或运算实现，并且把结果数除以该散列表大小。余数将是在 0 与散列表大小减 1 之间的一个数；这个余数表示散列表的索引。在步骤 1220，从分配表（ 225 ）确定类到服务器映射。在步骤 1230，把该映射信息发送给请求者。

图 13 描述请求者（ 203 ）逻辑的一个例子。在步骤 1310，请求者等待输入。在步骤 1315，对于一个目标请求，在步骤 1320 调用目标请求产生程序。该目标请求产生程序确定根据目标标识符（ 例如 URL ）将选择哪个服务器（ IP ）地址以便获得服务器的较好位置。将参照图 14 描述目标请求产生程序的一个详细例子。在步骤 1315，如果接收的输入不是一个目标请求，则处理进入到步骤 1350 。在步骤 1350，如果返回一个（ 以前请求的 ）目标，则在步骤 1360，检查该目标（ HTTP ）首部以便发现是否包括一个再分配标识符（ R-label ）。如果包括，则在步骤 1365，修改本地分配表（ 270 ）以便

反映类到服务器分配的改变。在步骤 1370，处理该接收的目标。在步骤 1325，如果返回一个（ 以前如在步骤 1440 请求的 ）映射请求，则在步骤 1330 将该（ 未决的 ）目标请求发送给指定的服务器。在步骤 1340，修改本地分配表（ 270 ）以便反映基于该映射请求的类到服务器的再分配。在步骤 1335，对于不是本发明焦点的其它类型的输入（ 例如一个推目标 ）则可以调用一个适当的杂项处理程序。

图 14 描述目标请求产生（ 267 ）逻辑的一个例子。如步骤 1410 所述，将该目标映射为它的目标类。在步骤 1420，如果从类到服务器分配表中得不到有关的服务器，则接着在步骤 1440，向仲裁器发送一个映射请求（ 因此延迟该目标请求直到如图 13 的步骤 1330 所述完成该映射请求时为止 ）。否则，在步骤 1430，该目标请求发送给由类到服务器分配表指定的服务器。

本领域那些技术人员将容易知道在本发明的精神和范围内可以使用对它的各种扩展。例如，在目标请求产生程序（ 步骤 1440 ），可以选择群中的一个仲裁服务器，而不发出一个映射请求。该映射请求表也可以包括关于每一个类到服务器映射的一个合理间隔。当该间隔结束时，可以发出一个映射请求（ 如在步骤 1440 ），以响应那个类中的下一个目标请求。

本领域那些技术人员还知道本发明可以适合于目标标识符到服务器的分级映射。例如，本发明可以结合常规的域名服务器（ DNS ）或如图 15 所述的基于 TCP 的路由工作。这里，该类到服务器分配表优选将每一个类（ 图 3 ）分配给一个虚拟的服务器。虚拟服务器的个数比服务器群中服务器的实际个数多。DNS（ 167 ）TCP 路由器接着可以将每个虚拟服务器动态映射为群中的一个实际服务器。

而且，通过由所请求的目标分段元信息在服务器处修改路由信息的原理也可以用于修改因特网的常规 DNS 路由。这与使用目标的 URL（ 或类 ）进行服务器分配以提高高速缓冲存储器命中率的特征无关。DNS 路由只是试图平衡具有相同信息的多个 Web 服务器间的负荷（ 参见例如，IBM 研究报告，RC20680，1997 年 1 月，Colajanni, M., Yu, P.,Dias,D.的“ 分布式 Web 服务器的调度算法 ” ）。常规的 DNS 具有关于每一个名称到地址映射的一个 TTL 周期。该映射被高速缓存在各种名称服务器。这可以导致当 DNS 用于群负荷平衡时，只具有有限

的控制。根据本发明，如果群中的一个服务器过载，可以以返回的目标（优选使用PICS标识符或同样的机制）将一个替换的服务器IP地址分段（没有额外的网络通信量）以便使业务流量改向到群中的另一个服务器并且因此提高负荷平衡。

在优选实施例中，DNS（167）收集从每个请求者发出的请求数并且将产生一个请求到服务器分配表以便平衡服务器间的负荷。（对于异质的服务器，所分配的负荷可以与服务器的处理能力成比例）。当一个（名称到地址）映射请求到达DNS（167）时，根据分配表中该请求者的名称（或IP地址）分配服务器（161...163）。该映射是分级的和多级的，例如，URL=>类=>虚拟服务器=>服务器。DNS可以根据比TTL小（得多）的测量间隔收集负荷统计并且修改分配表（225）。因此，能够很快产生一个新的分配表，以便更好地反映负荷状况。所有服务器（161...163）从DNS（167）得到分配表（225）的最新版本。因为以前，无需把改变通知请求者（110...153）；它们可以仍然根据以前的（名称到地址）映射发送请求。但是，如果一个服务器从一个请求者接收了不再分配给那个服务器的一个请求，则该服务器将通知将来的请求应该发给的那个服务器（161...163）的请求者。将仍然服务该当前请求并且可以例如使用PICS或类似机制用响应或返回的目标捎带新的分配信息。当一个服务器过载时，它可以向DNS（167）发送一个报警信号。每次接收一个报警信号时，DNS（167）能够重新计算分配表以便减少分配给任何过载服务器的请求者的数目。也可以把请求者划分成类以便该分配表然后成为一个类到服务器分配。

现在参照图15描述DNS（167）路由逻辑的一个例子。假设通过DNS（167）为请求者（110）分配一个服务器（162）。在现有技术中，这个映射对于一些TTL间隔，比如说5分钟是合理的。根据本发明，可以在一个更短的间隔，比如说1分钟产生一个修改的分配表，并且给请求者（110）分配了一个较少负荷的服务器（163）。请求者（110）不需要知道到目前为止的改变；它仍把下一个请求发送给同一服务器（162）。但是，服务器（162）已经从DNS（167）接收到新的分配表。服务器（162）将服务该请求，但是用一个返回的目标分段一个消息，以便告诉请求者（110）把将来的请

求发送给服务器（ **163** ）。这在不增加通信量的情况下消除了 **TTL** 的不利影响。

本领域的技术人员还知道本发明的动态路由方法也工作在一个不同类的请求者环境中，其中一些请求者是不知道该路由协议并且不参与提高高速缓冲存储器的命中率和负荷平衡合作的常规的代理/客户站。

既然已经描述了本发明的一个优选实施例，本领域的技术人员可以任选地进行替代、各种修改和改进。因此，应该知道该详细描只作为一个例子并不作为一种限制。本发明的适当范围由所附的权利要求书恰当限定。

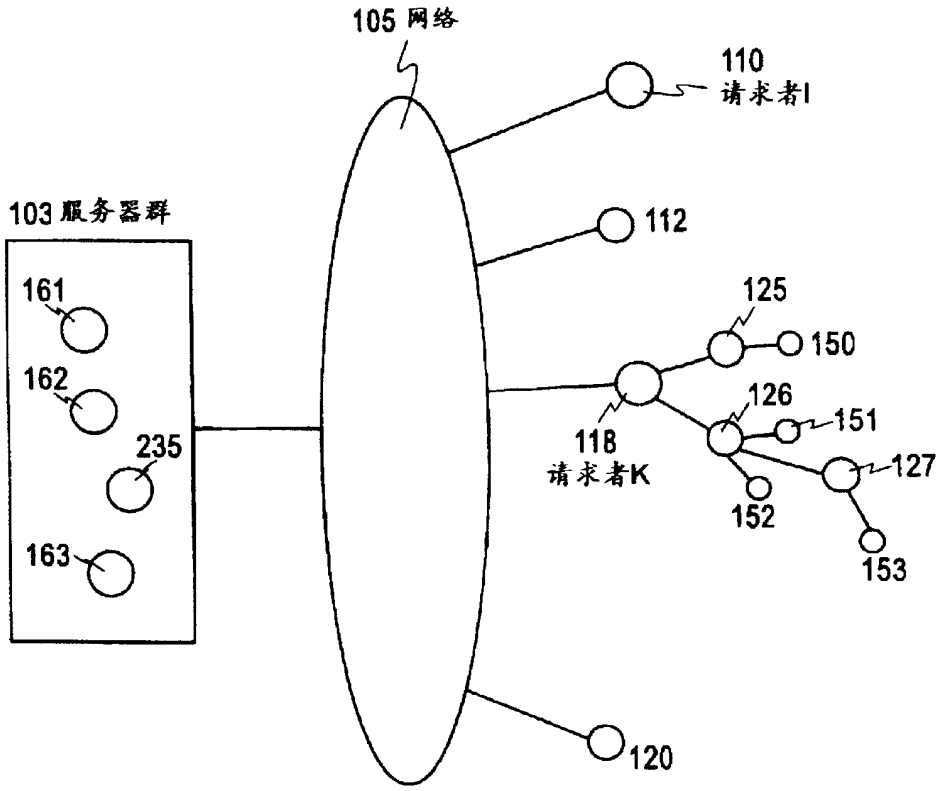

105 网络
110
请求者1
112
103 服务器群
161
162
235
163
125
150
126
151
127
118
请求者K
152
153
120

图 1

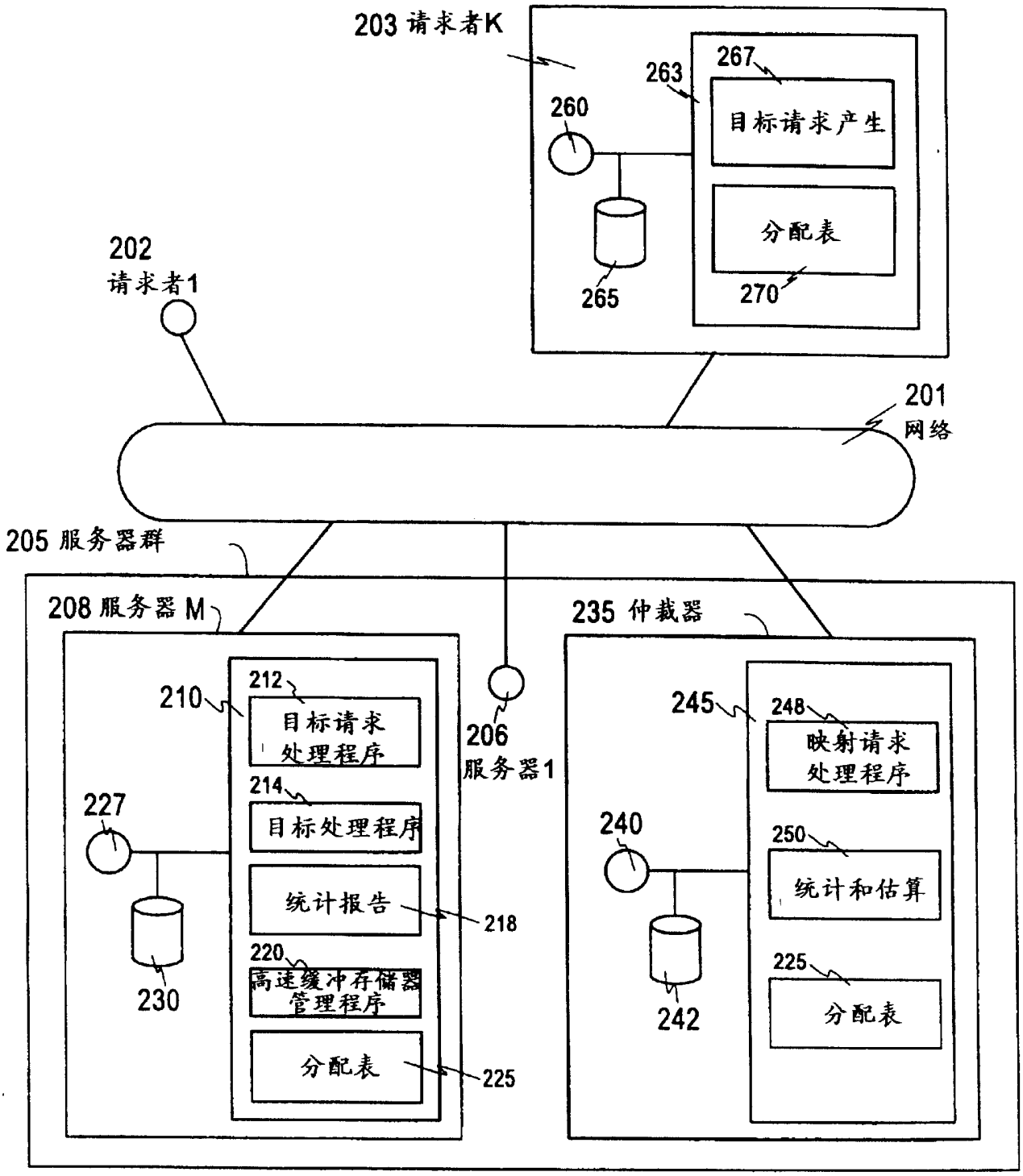

203 请求者K
267
263
260
目标请求产生
分配表
265
270
202
请求者1
201
网络
205 服务器群
208 服务器 M
235 仲裁器
210
212
目标请求
处理程序
214
目标处理程序
227
统计报告
218
220
高速缓冲存储器
管理程序
分配表
225
230
206
服务器1
245
248
映射请求
处理程序
240
250
统计和估算
225
分配表
242

图 2

分配表

N=16, M=3

类	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
服务器	2	3	2	2	2	2	2	3	1	3	1	2	2	3	2	3

图 3

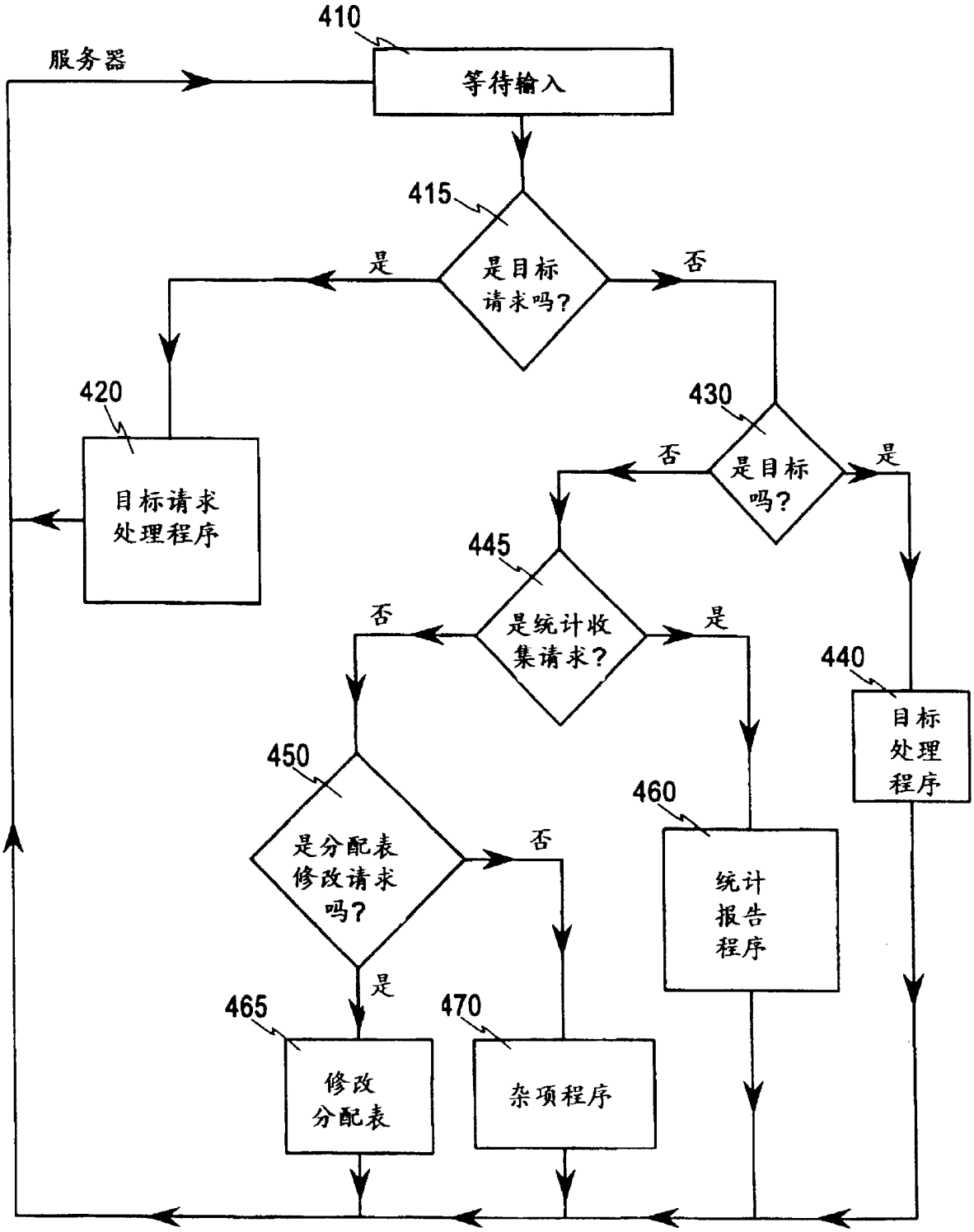

410
服务器
等待输入
415
是目标请求吗?
是
否
420
目标请求处理程序
430
是目标吗?
否
是
445
是统计收集请求?
否
是
440
目标处理程序
450
是分配表修改请求吗?
否
是
460
统计报告程序
465
修改分配表
470
杂项程序

图 4

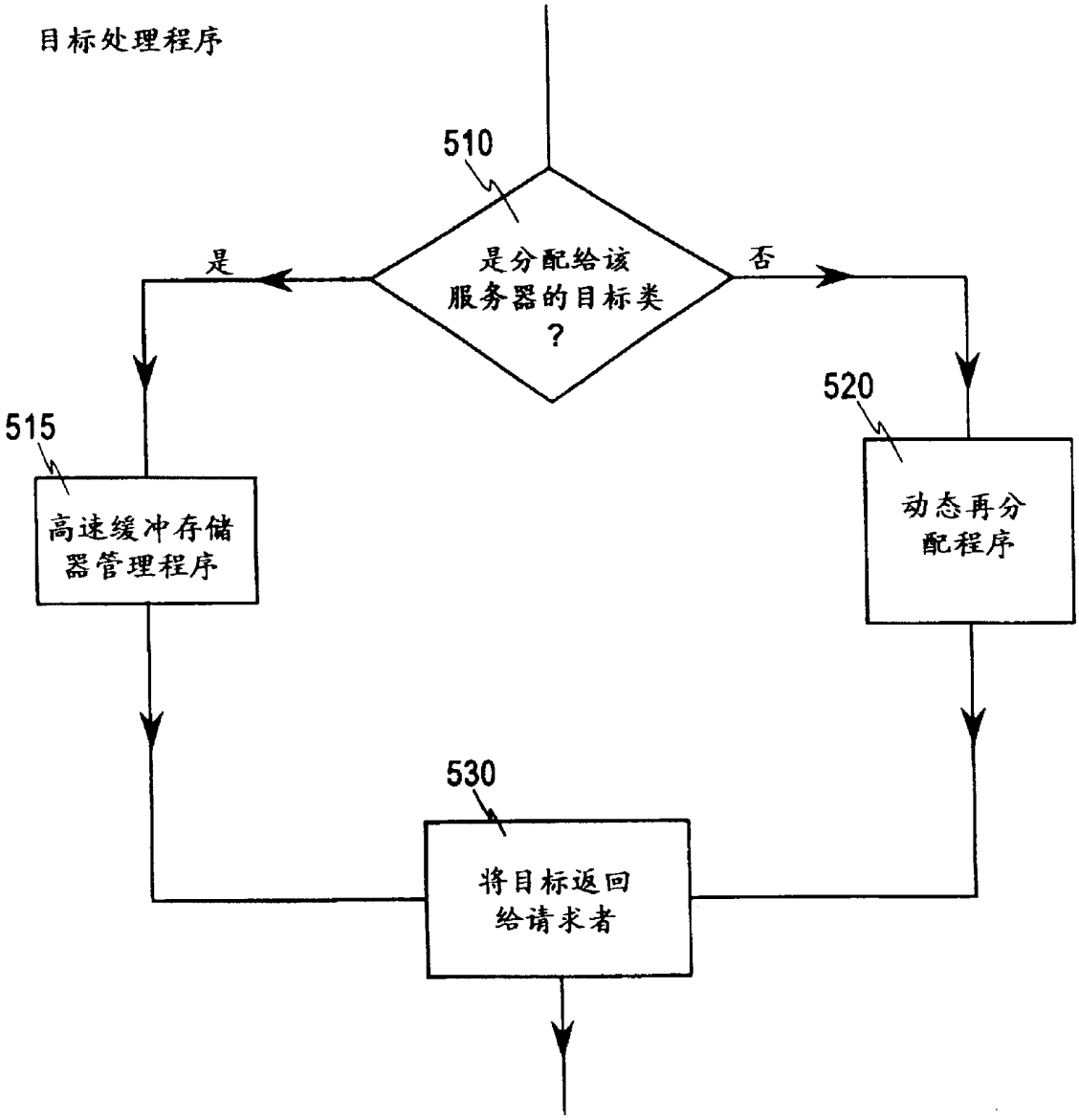

目标处理程序
510
是分配给该服务器的目标类？
是
否
515
高速缓冲存储器管理程序
520
动态再分配程序
530
将目标返回给请求者

图 5

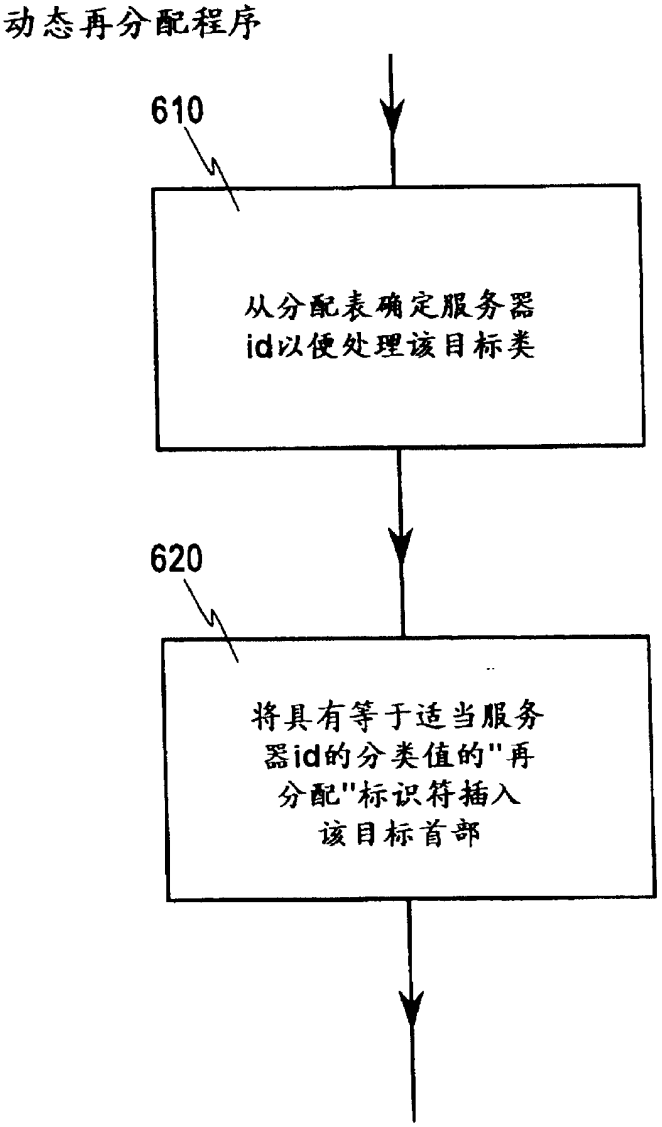

动态再分配程序
610
从分配表确定服务器id以便处理该目标类
620
将具有等于适当服务器id的分类值的"再分配"标识符插入该目标首部

图 6

目标请求处理程序

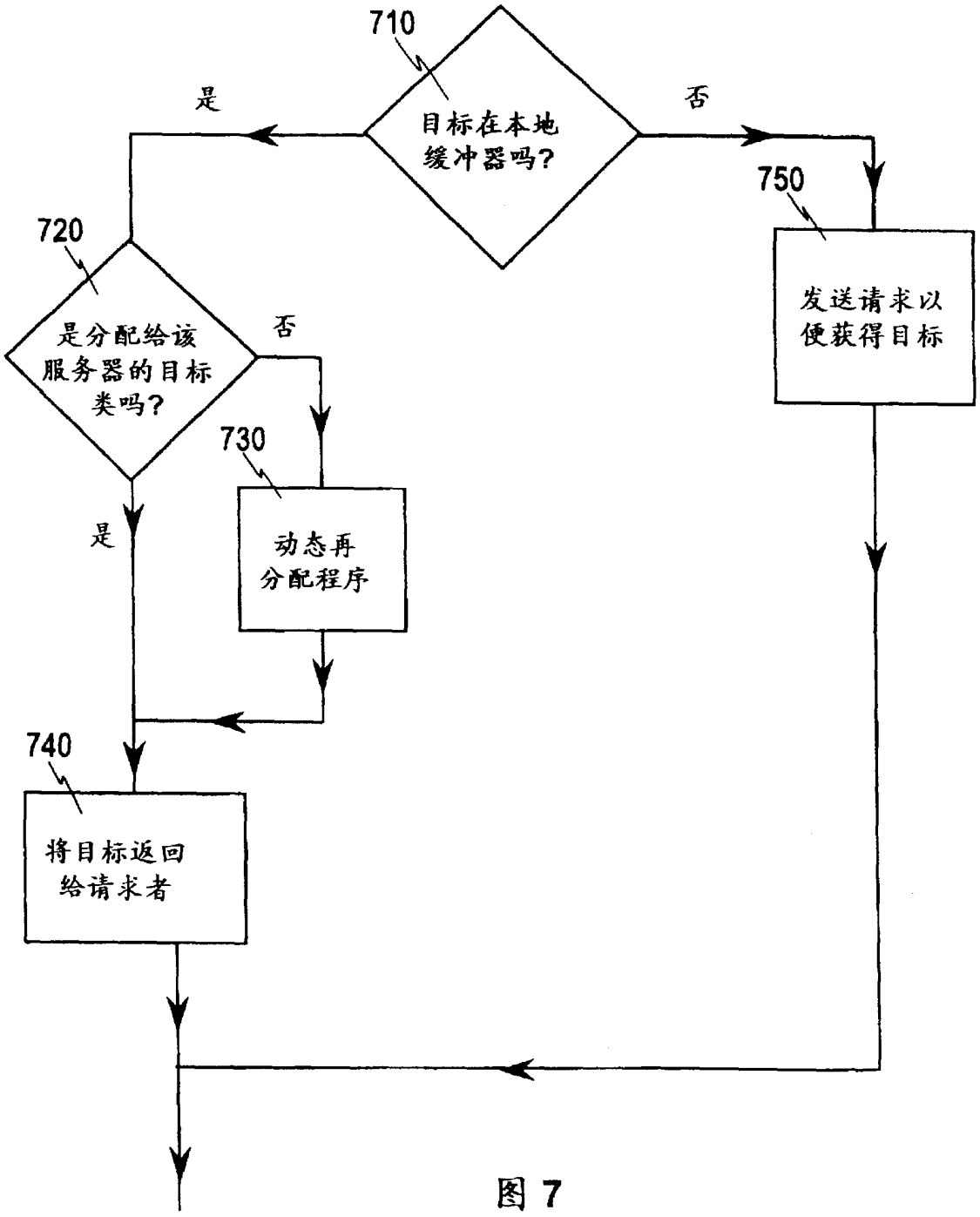

710
目标在本地缓冲器吗?
是
否
750
发送请求以便获得目标
720
是分配给该服务器的目标类吗?
否
730
动态再分配程序
是
740
将目标返回给请求者

图 7

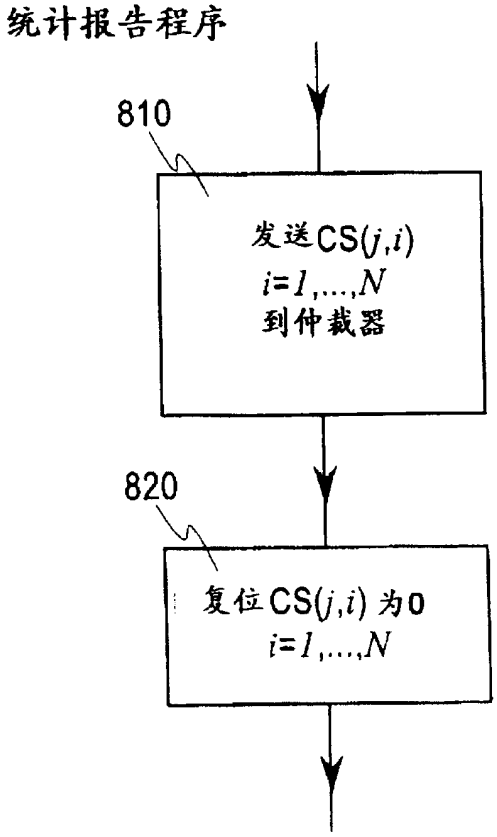


图 8

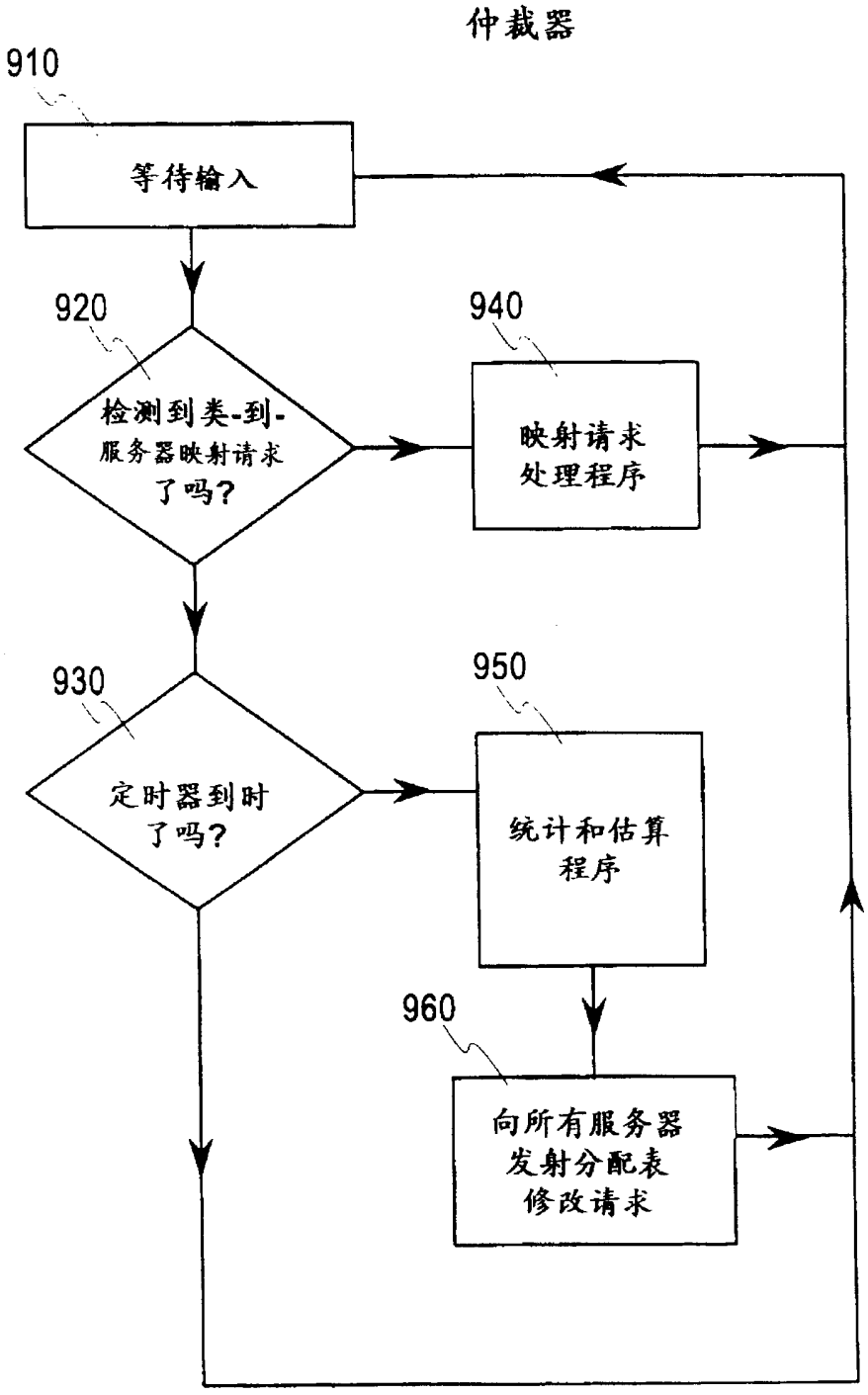

仲裁器
910
等待输入
920
检测到类-到-服务器映射请求了吗?
940
映射请求处理程序
930
定时器到时了吗?
950
统计和估算程序
960
向所有服务器发射分配表修改请求

图 9

统计和估算程序

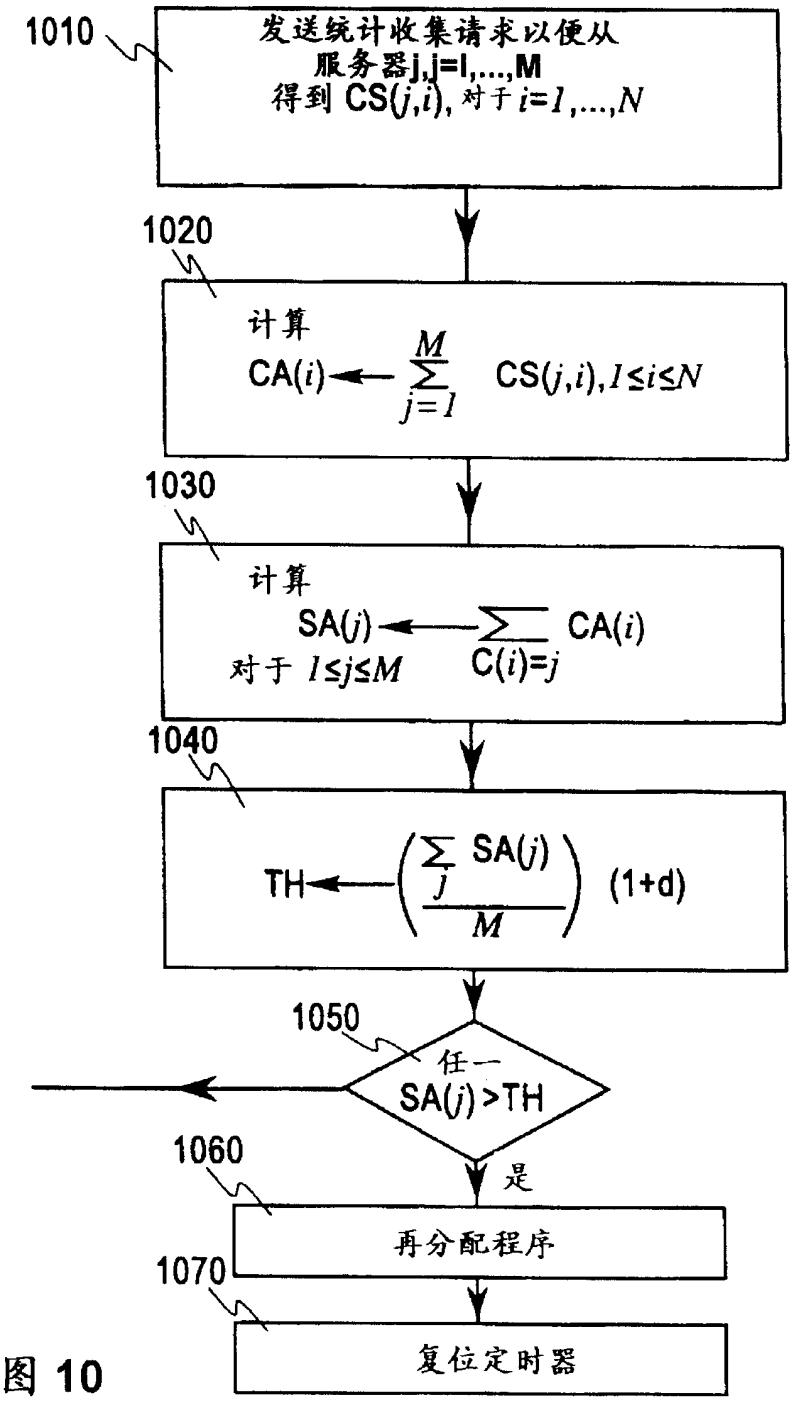

1010
发送统计收集请求以便从
服务器j,j=l,...,M
得到 CS(j,i), 对于 i=1,...,N
1020
计算
CA(i) ← Σ CS(j,i), 1≤i≤N
j=1
1030
计算
SA(j) ← Σ CA(i)
对于 1≤j≤M C(i)=j
1040
TH ← (Σ SA(j) / M) (1+d)
1050
任一
SA(j)>TH
是
1060
再分配程序
1070
复位定时器

图 10

再分配程序

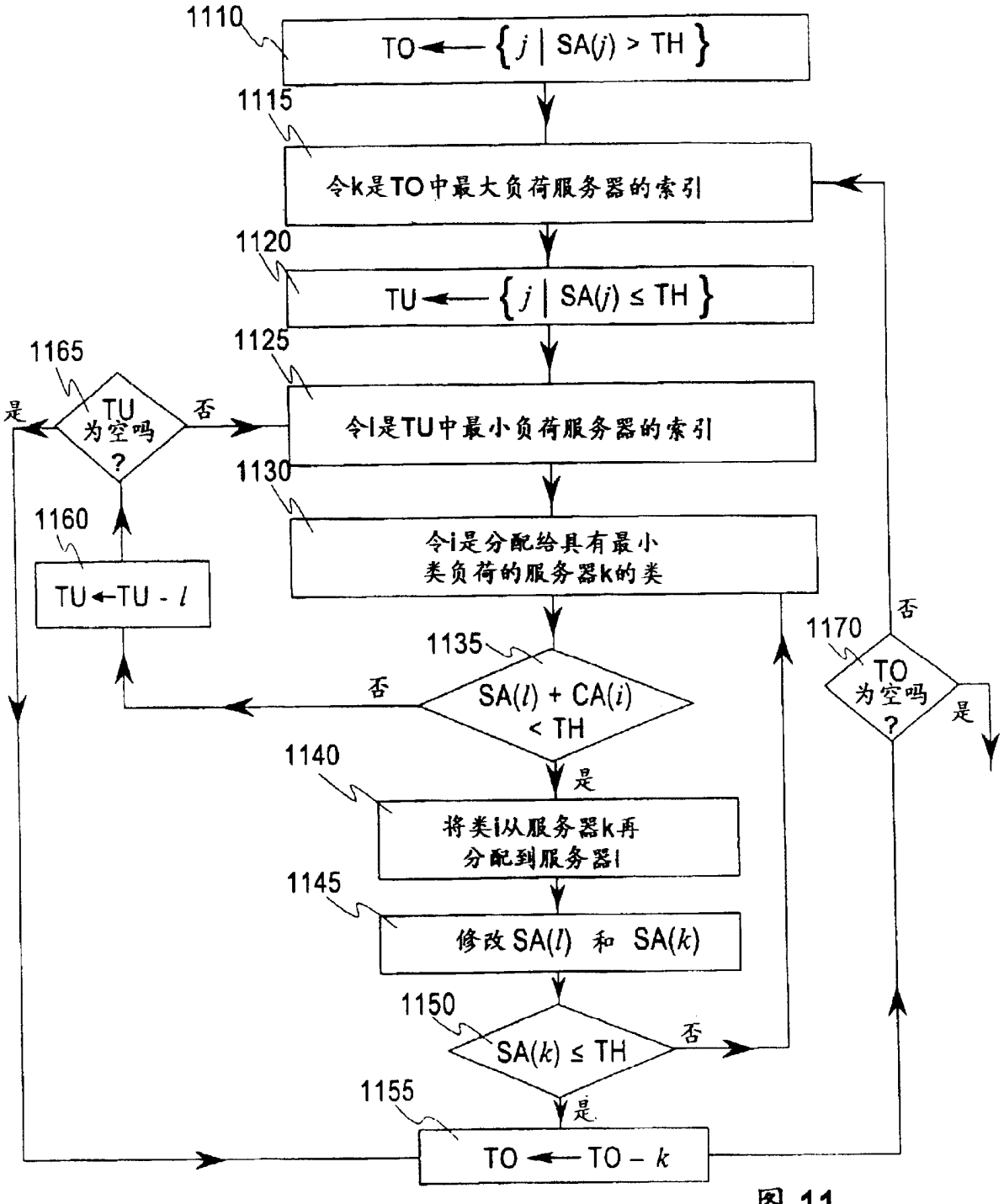

1110
TO ← { j | SA(j) > TH }
1115
令k是TO中最大负荷服务器的索引
1120
TU ← { j | SA(j) ≤ TH }
1125
令l是TU中最小负荷服务器的索引
1130
令i是分配给具有最小类负荷的服务器k的类
1135
SA(l) + CA(i) < TH
否
是
1140
将类i从服务器k再分配到服务器l
1145
修改 SA(l) 和 SA(k)
1150
SA(k) ≤ TH
否
是
1155
TO ← TO - k
1160
TU ← TU - l
1165
TU 为空吗 ?
是
否
1170
TO 为空吗 ?
否
是

图 11

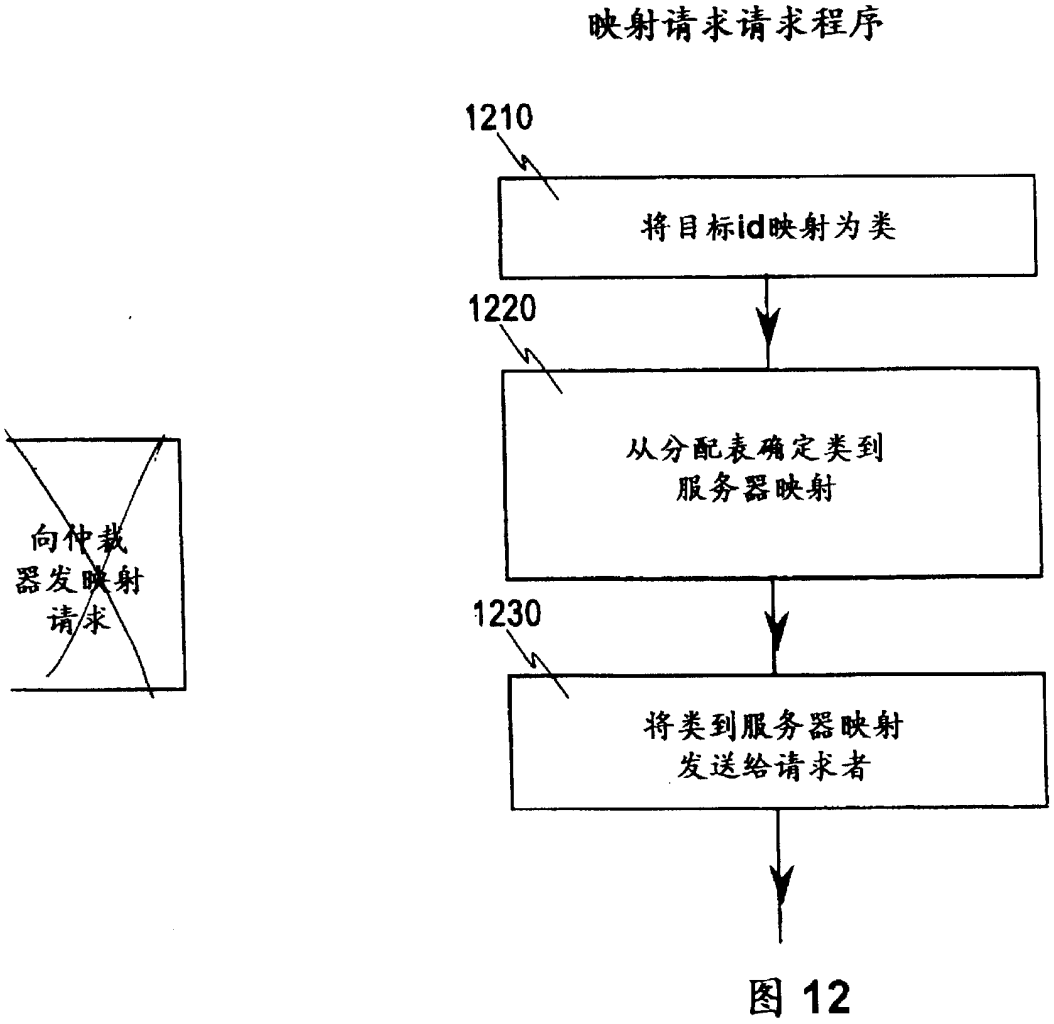

映射请求请求程序
1210
将目标id映射为类
1220
从分配表确定类到
服务器映射
1230
将类到服务器映射
发送给请求者
向仲裁
器发映射
请求

图 12

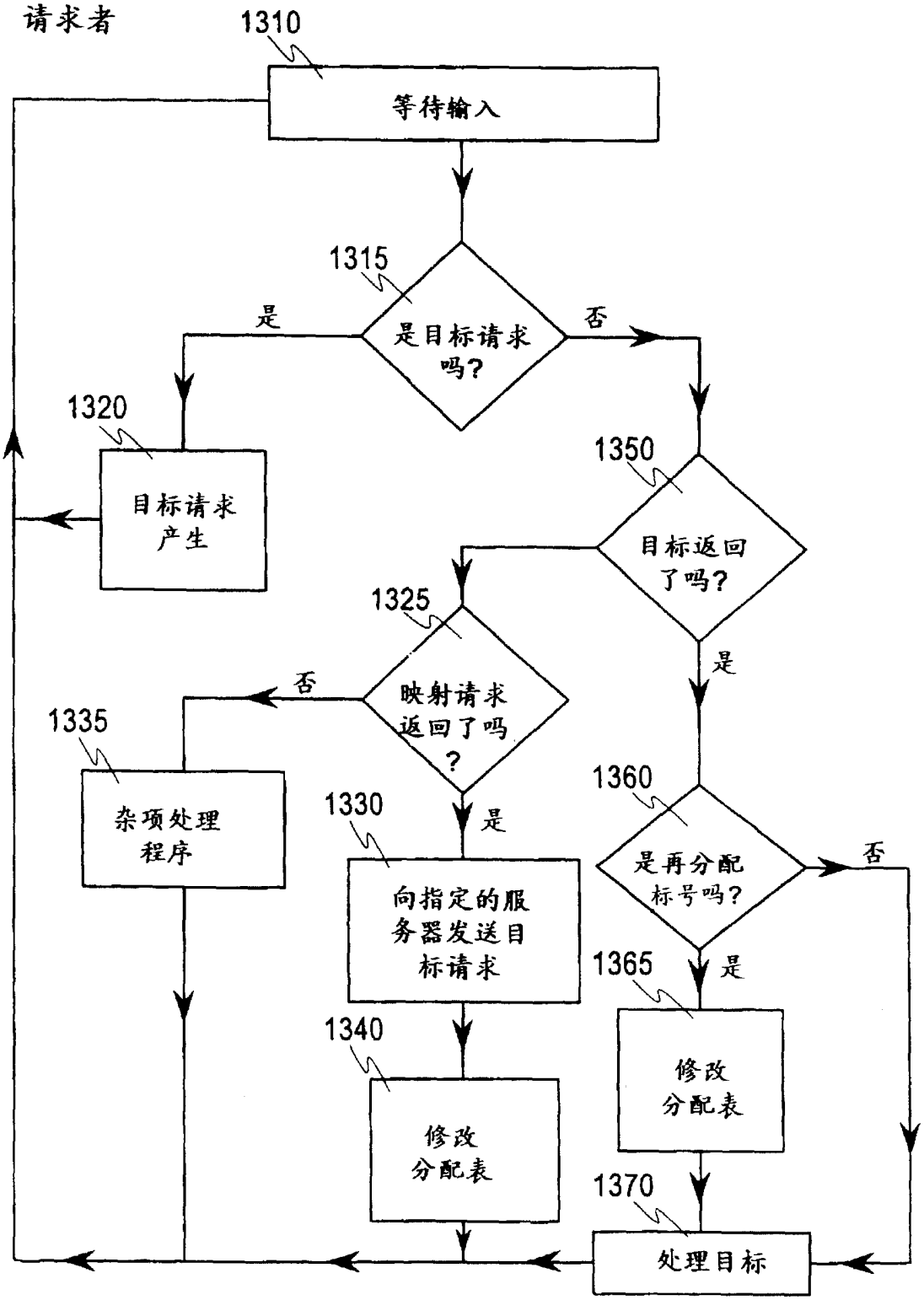

请求者
1310
等待输入
1315
是目标请求吗?
是
否
1320
目标请求产生
1350
目标返回了吗?
1325
映射请求返回了吗?
否
是
1335
杂项处理程序
1330
向指定的服务器发送目标请求
1340
修改分配表
1360
是再分配标号吗?
否
是
1365
修改分配表
1370
处理目标

图 13

目标请求产生

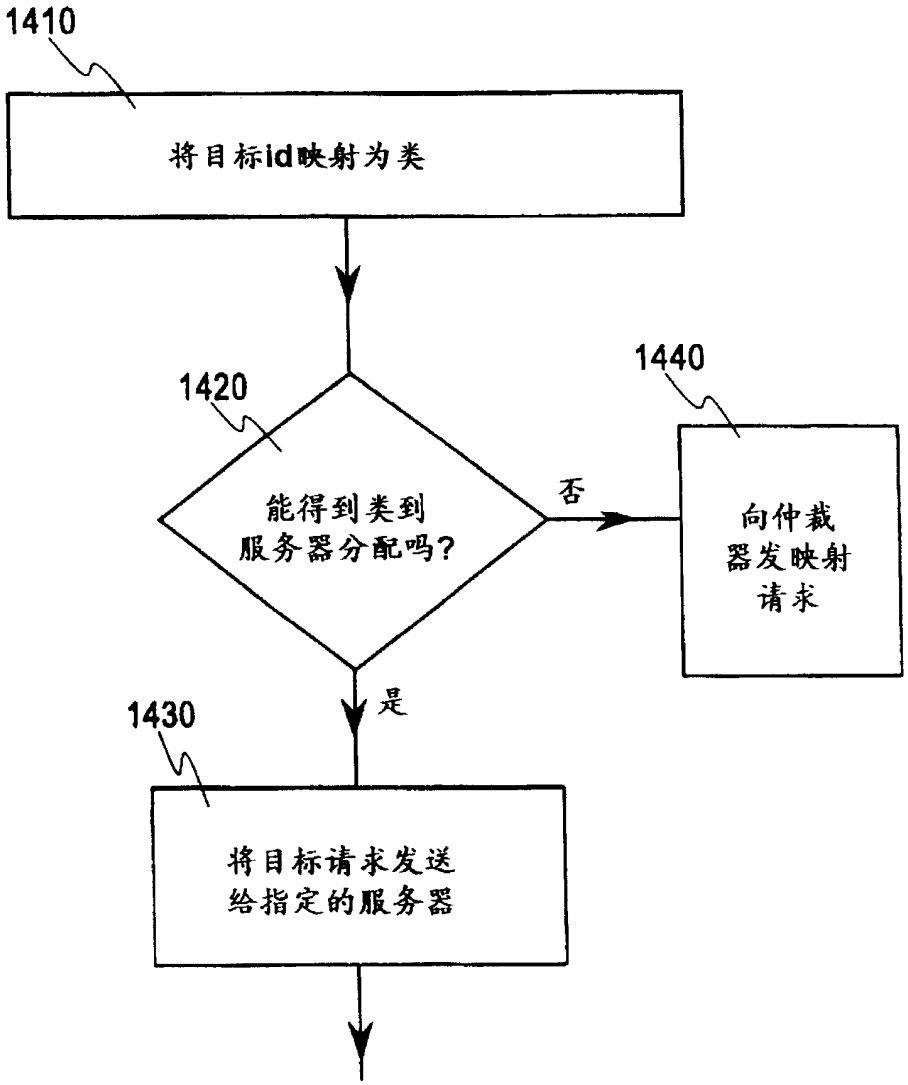

1410
将目标id映射为类
1420
能得到类到服务器分配吗?
否
1440
向仲裁器发映射请求
是
1430
将目标请求发送给指定的服务器

图 14

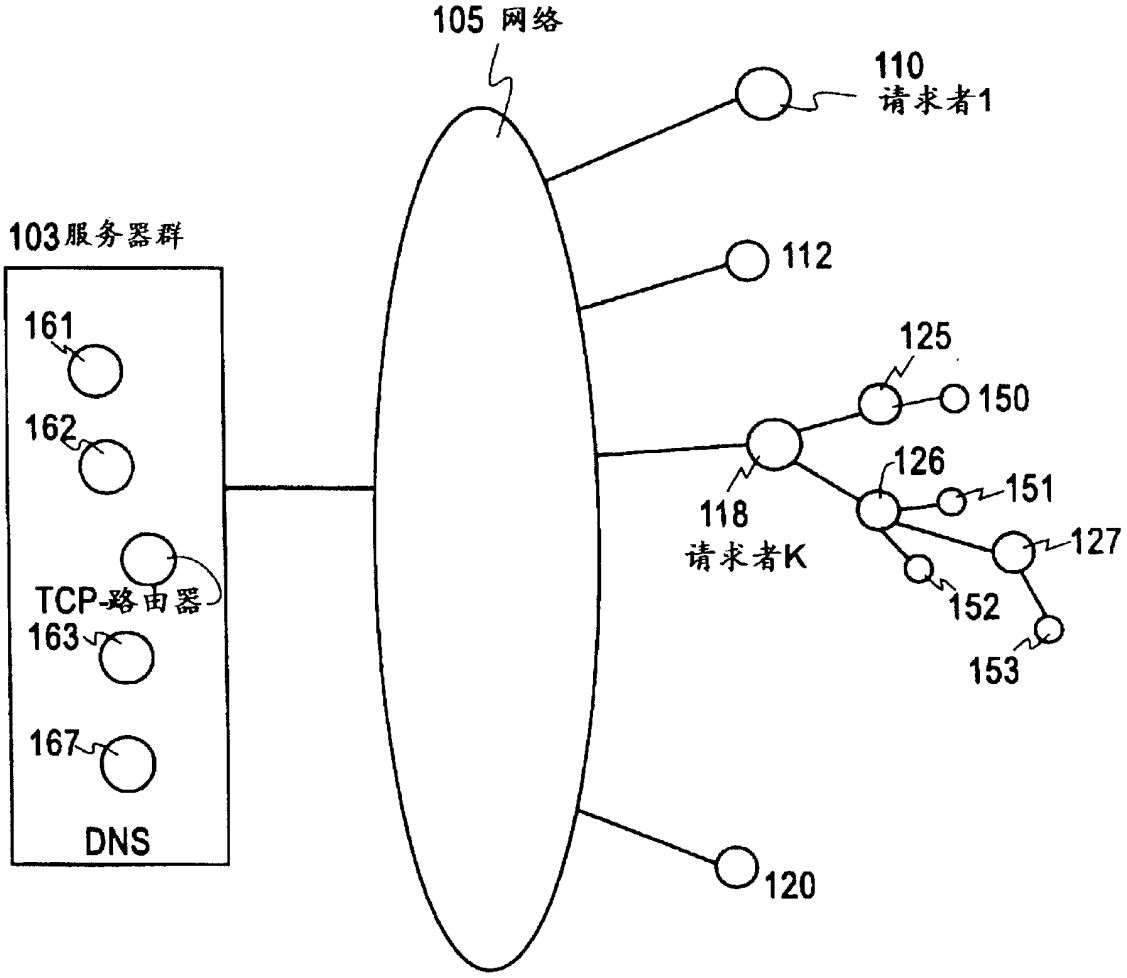

105 网络
110
请求者1
112
103服务器群
161
162
TCP-路由器
163
167
DNS
125
150
126
151
118
请求者K
127
152
153
120

图 15