[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 15/163 (2006.01)

[12] 发明专利说明书

专利号 ZL 02159493.7

[45] 授权公告日 2006 年 8 月 30 日

[11] 授权公告号 CN 1272724C

[22] 申请日 2002.12.31 [21] 申请号 02159493.7

[71] 专利权人 联想(北京)有限公司

地址 100083 北京市海淀区上地信息产业基地创业路 6 号

[72] 发明人 李电森 冯 锐 许正华 肖利民

审查员 俞立文

[74] 专利代理机构 北京同立钧成知识产权代理有限公司

代理人 刘 芳

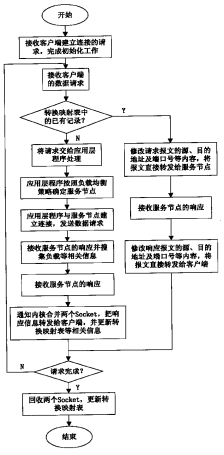
权利要求书 2 页 说明书 9 页 附图 4 页

[54] 发明名称

基于内核中套接字对接的第七层负载均衡的方法

[57] 摘要

一种基于内核中套接字对接的第七层负载均衡的方法，前置节点接收客户端请求；修改请求报文源、目的地址和端口号，将报文直接发送给服务节点；接收服务节点的响应；修改响应报文的源、目的地址和端口号，将报文直接发送给服务节点；如果请求完成，则回收两个套接字，更新转换映射表；否则继续接收客户端的请求。 本发明可充分利用传输层转发和应用层转发的优点，既能提高报文转发效率，又具有很大的灵活性，从而更快、更均衡地把客户端的请求分发到服务节点上；并且可用该设计通用的中间件，实现所有端口上的报文转发功能；也可对报文进行复杂的分析，应用特殊的过滤规则，实现功能复杂的防火墙；甚至可以对现有 TCP 连接动态地进行迁移。



1、一种基于内核中套接字对接的第七层负载均衡的方法，其特征在于：至少包括：

步骤 10：前置节点接收到客户端的请求；

步骤 20：修改请求报文的源、目的地址和端口号，将报文直接发送给服务节点；

步骤 30：接收服务节点的响应；

步骤 40：修改响应报文的源、目的地址和端口号，将报文直接发送给客户端；

步骤 50：如果请求完成，则回收两个套接字，更新转换映射表；否则执行步骤 10。

2、根据权利要求 1 所述的基于内核中套接字对接的第七层负载均衡的方法，其特征在于：步骤 20 之前还进一步包括：

步骤 11：如果前置节点第一次接收到客户端的请求，则执行步骤 12，否则，执行所述步骤 20；

步骤 12：将该请求数据从内核空间拷贝到用户空间，交由应用层程序处理，应用层程序对请求数据进行分析，并根据负载均衡策略和服务节点的状态，将本次服务请求交由相应的服务节点处理；

步骤 13：应用层程序和选定的服务节点建立套接字连接，将用户请求数据重新封装，并将其转发给选定的服务节点；

步骤 14：服务节点对用户请求进行处理，并将响应报文发给前置节点；

步骤 15：前置节点的应用层程序接收到响应报文，构建转换映射表，通知内核合并该两个套接字，同时放弃对两个套接字的控制权，执行所述步骤 50。

3、根据权利要求 2 所述的基于内核中套接字对接的第七层负载均衡的方法，其特征在于：所述的负载均衡策略至少为：轮询算法；或加权的轮询算法；或最少连接数算法；或加权的最少连接数算法；或基于请求位置的最少连接数

算法，即：将相同客户端 IP 的请求发送给相同的服务节点处理；或预先分配任务法，即：根据承担任务情况来选择负载最轻的节点；或加权的预先分配任务法，即：根据承担任务情况及节点的性能来综合选择负载最轻的节点；或客户端 IP 地址分区法，即：将不同客户端的 IP 地址分为多个区，来自一个区的请求全部被分发到一个指定的节点。

基于内核中套接字对接的第七层负载均衡的方法

技术领域

本发明涉及机群中负载均衡的方法，尤其涉及一种基于内核中套接字对接的第七层负载均衡的方法，属于计算机网络技术领域。

背景技术

随着计算机网络应用的普及，计算机网络用户的数量在不断膨胀，这对于服务提供商提出了更大的挑战：按照传统的方法使用单台服务器提供服务已经远远不能满足庞大的用户请求的需求；现在通常都会使用多台服务器向用户提供服务。但是，如何才能把所有的用户请求均衡地分布到后台服务器上，则是负载均衡需要解决的问题。

如果按照实施负载均衡策略组件所处的位置来划分，可以将负载均衡分为两类：基于域名服务器（Domain Name Server，简称 DNS）的负载均衡和基于服务端应用层与/或网际协议（Internet Protocol，简称 IP）层的负载均衡。基于 DNS 的负载均衡通常使用 DNS 轮询（round-robin）机制，在 DNS 上为一个名字设置多个 IP 地址，分别指向后台实际提供服务的服务器。其机制十分简单，却缺乏灵活性，而且有很多限制条件；例如：后台的服务器必须使用外部地址，能和客户端直接进行通信。而基于服务端应用层与/或 IP 层的负载均衡则不同，它可以根据一定的策略，对请求数据的具体内容进行分析，适当修改请求报文的内容，将其发送到适当的服务节点；例如：专用的负载均衡器硬件、环球网（Web）服务器提供的万维网（World Wide Web，简称 WWW）服务程序所用的超文本传输协议（Hypertext Transfer Protocol，简称 HTTP）重定向功能等，都可以在服务端根据应用的请求而实施负载均衡。

集群技术可以把多台服务器封装成一些服务器聚集，统一对外界提供服

务。这种技术具有很多优点，例如：可扩展性好、对外界屏蔽内部细节等。在集群中通常会为用户提供一个唯一的单一登录点（称为前置节点），它和集群中的所有服务器保持连通，这种特点决定了可以在集群中实施特殊的负载均衡机制。

集群中的前置节点通常就是为用户提供的单一登录点，它负责转发客户端和服务器之间交互的数据，并根据一定的策略，将客户端的请求均衡地分发到各个服务节点上，并将服务节点的响应转发给客户端，从而透明地为用户提供强大的服务处理能力。

实际上，前置节点的主要任务有两个：报文转发和均衡负载。报文转发可以在两个层次上进行：传输层和应用层。基于传输层的报文转发可以使用网络地址转换（NAT）、目标地址转换、IP 封装、直接包转发等技术；这样可以直接修改 IP 数据报的源地址、目的地址，或者另外附加一个 IP 数据报首部，或者直接修改数据报的以太网地址，最终将报文发送到目的地址。

由于上述的过程只需修改传输控制协议（Transfer Control Protocol，简称 TCP）首部或 IP 首部，不涉及传输数据的变化，因此，能够快速地转发报文；其缺点是缺乏灵活性。而基于应用层的包转发方式则不同，它把接收到的数据报逐步解析，剥离 IP 首部和 TCP 首部，最终得到应用数据；然后根据应用数据选择适当的目的地址，重新封装 TCP 首部和 IP 首部，通过网络驱动程序转发给内部网络的目的地址。其优点是可以根据请求的内容选择目标节点，并能对请求数据进行缓冲，同时可以屏蔽内部节点的细节和故障，对终端用户透明。由于应用层是开放式系统互联参考模型（Open System Interconnect Reference Model 简称 OSI 模型）中的第七层，因此这种报文转发技术通常就称为第七层转发。

由于前置节点需要处理各种用户请求，这涉及到报文分析、数据库查询等大量操作，不可能完全在内核空间中完成，因此，前置节点一般都采用应用层程序来实现报文转发和负载均衡。

以用于电子邮件接收的（Point of Presence，also Post Office Protocol，简称 POP3）服务为例：如果只对外提供一台 POP3 服务器，而且可以直接与客户端连通，那么情况就非常简单：客户端只需要和 POP3 服务器的 110 端口建立一条 TCP 连接，并向服务器请求数据即可。而对于一个提供 POP3 服务的集群来说，情况就变得非常复杂：通常大型的邮件服务器中的用户都是分布式的，用户的数据文件可能分布在多台邮件服务器上，而客户端对此并不知情。假设客户端需要读取“username@domain.com”的邮件，过程如图 1 所示：

详细步骤如下：

1、客户端向前置节点（domain.com）的 110 端口发起请求，要求读取用户“username@domain.com”的邮件；

2、前置节点的网卡接收到用户请求之后，操作系统将其从内核空间拷贝到用户空间，交给应用层的程序处理；

3、应用层程序对用户请求进行分析，确定是 POP3 请求，需要查询邮件用户数据库 LDAP/DB；

4、返回查询结果，确定用户“username”的数据文件保存在邮件服务器 Mail Server1 上；

5、将用户空间中的数据拷贝到内核空间，重新封装报文；

6、和邮件服务器 Mail Server1 建立套接字（socket）连接，发送请求数据；

7、邮件服务器 Mail Server1 接收到请求之后，读取用户“username”的邮件数据，作为响应报文发送回前置节点；响应报文沿着来路反向返回，直至发送到客户端。

如果客户端的请求是新建立一个“username@domain.com”的用户，那么上面的步骤 3 就不再是简单地查询邮件用户数据库；应用层程序会搜集所有邮件服务器的负载、磁盘空间等信息，采用相应的策略，将该用户的请求分配到最合适的一个邮件服务器上，并在这台邮件服务器上为该用户建立数据文件，

并将结果返回给前置节点，由前置节点转发给客户端，这样就能保证所有的邮件服务器之间的负载是均衡的。

TCP 是面向连接的协议，普通的 TCP 连接的建立需要经历一个三次握手的过程:

首先，请求端（通常称为客户端）向连接端（服务器）发送一个 SYN 报文，指明要打开的端口，报文中包括客户端的初始序号 ISNC; 接着，服务器返回一个命令正确应答（ACK）报文进行应答，其中的应答号等于 ISNC+1，同时向客户端发送一个 SYN 报文，其中包括服务器的初始序号 ISNS; 最后，客户端将 ISNS+1 作为应答号向服务器发送 ACK 报文进行应答，同时可以发送请求数据报文。

此后，客户端和服务器之间就使用这个 TCP 连接进行交互数据，构成一个完整的会话。

可以使用源 IP 地址、源端口号、目的 IP 地址以及目的端口号来唯一标识一个会话; 而每次会话的具体过程还需要另外两个值进行标识: 序列号（SEQ）和应答号（ACK）。其中源 IP 地址和目的 IP 地址位于 IP 数据报的首部中，而源端口号、目的端口号、序号、应答号则位于 TCP 报文的首部中。

和基于传输层的报文转发技术相比，第七层转发有一个致命的缺点就是效率非常低，因为它将传输的应用数据从内核空间拷贝到用户空间，未对其进行任何修改，又拷贝回内核空间，而且还引起了相应的上下文切换的开销。FreeBSD 中提供了一种转移套接字（divert socket）机制，允许在用户层直接处理 TCP/用户数据报协议（User Datagram Protocol，简称 UDP）报文，可以简化应用层转发报文的过程，但是仍然不可避免应用数据进出内核空间的开销。

实际上，转发节点和客户端、服务器分别建立的两个 TCP 连接并非毫不相干。这两个连接一旦建立，在内核空间中就可以得到这两个连接各自的源地址、源端口号、目的地址和目的端口号; 而且请求和应答的序列号之间也存在

一定的对应关系。分别以符号串 SEQ_{C-D}、SEQ_{D-S}、SEQ_{S-D}、SEQ_{D-C} 表示客户端到转发节点、转发节点到服务器、服务器到转发节点、转发节点到客户端之间报文中的序列号，以 ACK_{C-D}、ACK_{D-S}、ACK_{S-D}、ACK_{D-C} 表示客户端到转发节点、转发节点到服务器、服务器到转发节点、转发节点到客户端之间报文中的应答序列号，并设：

$$\Delta_R = SEQ_{D-S} - SEQ_{C-D} = ACK_{D-S} - ACK_{C-D}$$
$$\Delta_A = SEQ_{D-C} - SEQ_{S-D} = ACK_{D-C} - ACK_{S-D}$$

转发节点在第一次对客户端和服务器之间的请求/响应信息进行转发之后，就可以计算出这两个值，而且这两个值在该会话的生存期内保持不变；此后转发节点就可以根据接收到的客户端报文中的 SEQ_{C-D} 和服务器报文中的 SEQ_{S-D} 来计算出对应的 SEQ_{D-S} 和 SEQ_{D-C}。

图 2 所示为使用基于内核套接字对接机制对报文进行转发的示意图，客户端经由转发节点完成一次请求任务的过程为：

1、客户端向转发节点发送第一个 SYN 报文，其中包括 32 位的客户端 IP 地址（简称 SA）、32 位的转发节点的 IP 地址（简称 DA）、16 位的客户端端口号（简称 SP）、16 位的转发节点端口号（简称 DP）、初始序列号（简称 ISNC）；

2、转发节点在传输层接收到 SYN 报文后，产生一个记录，记下 SA、DA、SP、DP、ISNC 等值，并将该报文传递给上层等待接收的应用层程序。应用层程序产生一个初始序列号（简称 ISND），并向客户端发送应答连接包；其中：ACK=ISNC+1、SEQ=ISND；

3、客户端向转发节点开始请求数据，SEQ=ISNC+1，ACK=ISND+1；

4、转发节点上的应用层程序接收到数据后，对请求数据的内容进行分析，确定目标服务器节点；

5、转发节点重复 1 到 3 步，和真正提供服务的服务节点建立连接，并将请求数据发送给服务节点；

6、服务节点接收到请求之后，将响应数据发送给转发节点；

7、转发节点接收到服务节点的响应数据之后，将响应数据重新组装，将其发送给客户端；同时查找转换映射表中的数据，计算相应的转发前后的序号差值Δ_R和Δ_A，完成转换映射表的构造；

8、转发节点的应用层程序确定可以进行双向转发之后，通过输入输出控制（ioctl）系统调用通知内核合并两个 socket，同时释放对这两个 socket 的控制权；

9、以后的数据转发，根据图 2 所示的机制，在内核中修改 TCP 报文首部、IP 数据报首部中相应的内容，直接对数据进行转发，所转发的数据将不再进出用户空间，因此将极大提高转发的效率；

10、连接的双方如果一方中断，则释放整个连接。

发明内容

本发明的主要目的是提供一种基于内核中套接字对接的第七层负载均衡的方法，可以减少前置节点/服务器在转发数据时由于数据在内核空间和用户空间之间拷贝，以及相应的上下文切换等所引起的系统开销，降低前置节点的负载，缩短用户请求的响应时间。

本发明的目的是这样实现的：

一种基于内核中套接字对接的第七层负载均衡的方法，至少包括：

步骤 10：前置节点接收到客户端的请求；

步骤 20：修改请求报文的源、目的地址和端口号，将报文直接发送给服务节点；

步骤 30：接收服务节点的响应；

步骤 40：修改响应报文的源、目的地址和端口号，将报文直接发送给客户端；

步骤 50：如果请求完成，则回收两个套接字，更新转换映射表；否则执行步骤 10。

在上述的步骤 20 之前还进一步包括:

步骤 11: 如果前置节点第一次接收到客户端的请求，则执行步骤 12，否则，执行所述步骤 20;

步骤 12: 将该请求数据从内核空间拷贝到用户空间，交由应用层程序处理，应用层程序对请求数据进行分析，并根据负载均衡策略和服务节点的状态，将本次服务请求交由相应的服务节点处理;

步骤 13: 应用层程序和选定的服务节点建立套接字连接，将用户请求数据重新封装，并将其转发给选定的服务节点;

步骤 14: 服务节点对用户请求进行处理，并将响应报文发给前置节点;

步骤 15: 前置节点的应用层程序接收到响应报文，构建转换映射表，通知内核合并该两个套接字，同时放弃对两个套接字的控制权，执行所述步骤 50。

步骤 12 中所述的负载均衡策略至少为:

轮询算法;

或加权的轮询算法;

或最少连接数算法;

或加权的最少连接数算法;

或基于请求位置的最少连接数算法，即: 将相同客户端 IP 的请求发送给相同的服务节点处理;

或预先分配任务法，即: 根据承担任务情况来选择负载最轻的节点;

或加权的预先分配任务法，即: 根据承担任务情况及节点的性能来综合选择负载最轻的节点;

或客户端 IP 地址分区法，即: 将不同客户端的 IP 地址分为多个区，来自一个区的请求全部被分发到一个指定的节点。

本发明所提供的基于内核中套接字对接的第七层负载均衡的方法，可以充分利用传输层转发和应用层转发的优点，既能提高报文转发效率，又具有很大的灵活性，从而更快、更均衡地把客户端的请求分发到服务节点上。

并且可以使用这种机制设计一种通用的中间件，实现所有端口上的报文转发功能；也可以对报文进行复杂的分析，应用特殊的过滤规则，实现功能复杂的防火墙；甚至可以对现有 TCP 连接动态地进行迁移。

附图说明

图 1 为现有技术在集群中使用应用层程序处理 POP3 服务请求的过程；

图 2 为本发明使用基于内核套接字对接机制对报文进行转发的示意图；

图 3 为使用基于内核中套接字对接技术实现第七层负载均衡的流程图；

图 4 为套接字对接之后的通信过程的流程图。

具体实施方式

以下结合具体的实施例对本发明做进一步的说明：

参见图 3，本发明的方法首先接受客户端建立连接的请求，完成初始化的工作，然后就接收客户端的数据请求；根据该请求判断转换映射表中是否已有相应的记录，如果有记录，则修改请求报文的源地址、目的地址及端口号等内容，将报文直接转发给服务节点；然后接收服务节点的响应，接着修改响应报文的源、目的地址及端口号等内容，将报文直接转发给客户端；再进一步判断请求是否完成；若未完成，则返回接收客户端的数据请求，继续循环，接受客户端建立连接的请求；若已完成，则回收两个 Socket，更新转换映射表，结束。

如果上述的转换映射表中没有记录，则将请求交给应用层程序处理，应用层程序按照负载均衡策略确定服务节点，然后应用层程序与服务节点建立连接，发送数据请求；接着接收服务节点的响应并搜集负载等相关信息，接收到服务节点的响应后，通知内核合并两个 Socket，把响应信息转发给客户端，再判断请求是否完成，若未完成，则返回接收客户端的数据请求，继续循环；若已完成，则回收两个 Socket，更新转换映射表，结束。

由图 3 可知，前置节点的应用程序在接收到邮件服务器 Mail Server1 的

响应报文之后，就可以确定客户端和服务节点之间可以相互转发报文，因此通知内核将两个 socket 合并，此后前置节点在转发客户端和邮件服务器 Mail Server1 之间的通信时，就可以直接在内核空间完成，在两个 socket 合并之后，客户端和服务节点之间的通信过程就如图 4 所示，因此，大大减少了传输时间，极大地提高了效率。

最后所应说明的是：以上实施例仅用以说明而非限制本发明的技术方案，尽管参照上述实施例对本发明进行了详细说明，本领域的普通技术人员应当理解：依然可以对本发明进行修改或者等同替换，而不脱离本发明的精神和范围的任何修改或局部替换，其均应涵盖在本发明的权利要求范围当中。

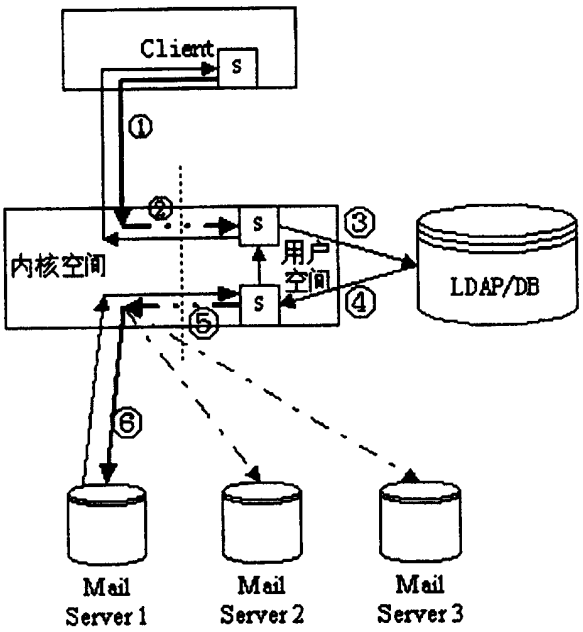

Client
S
①
②
S
③
内核空间
用户空间
LDAP/DB
S
④
⑤
⑥
Mail Server 1
Mail Server 2
Mail Server 3

图 1

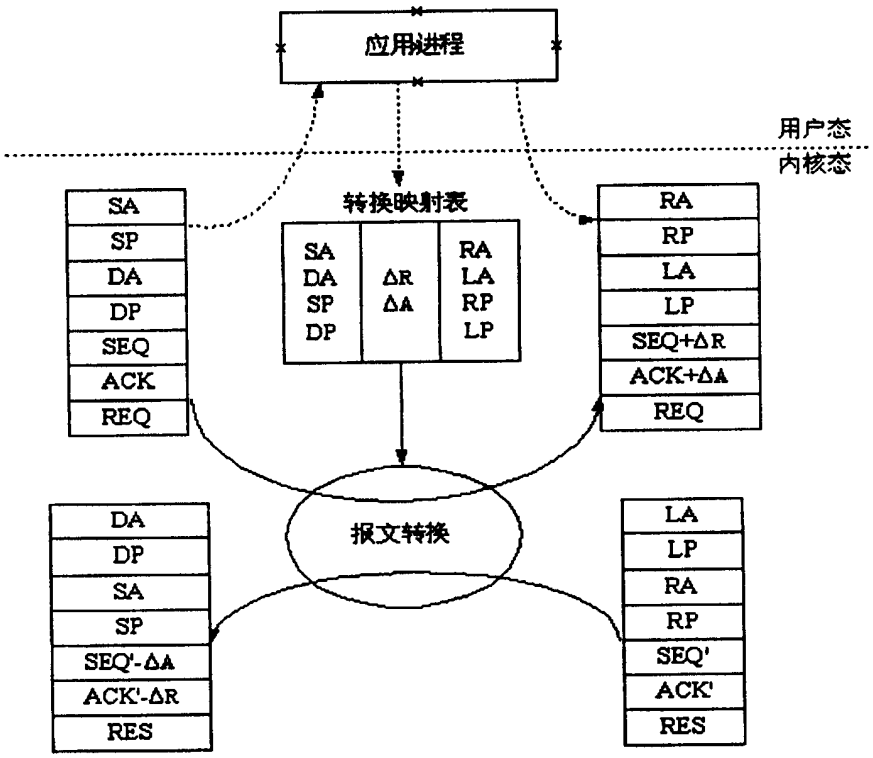

应用进程
用户态
内核态
转换映射表
SA
SP
DA
DP
SEQ
ACK
REQ
SA
DA
SP
DP
ΔR
ΔA
RA
LA
RP
LP
RA
RP
LA
LP
SEQ+ΔR
ACK+ΔA
REQ
报文转换
DA
DP
SA
SP
SEQ'-ΔA
ACK'-ΔR
RES
LA
LP
RA
RP
SEQ'
ACK'
RES

图 2

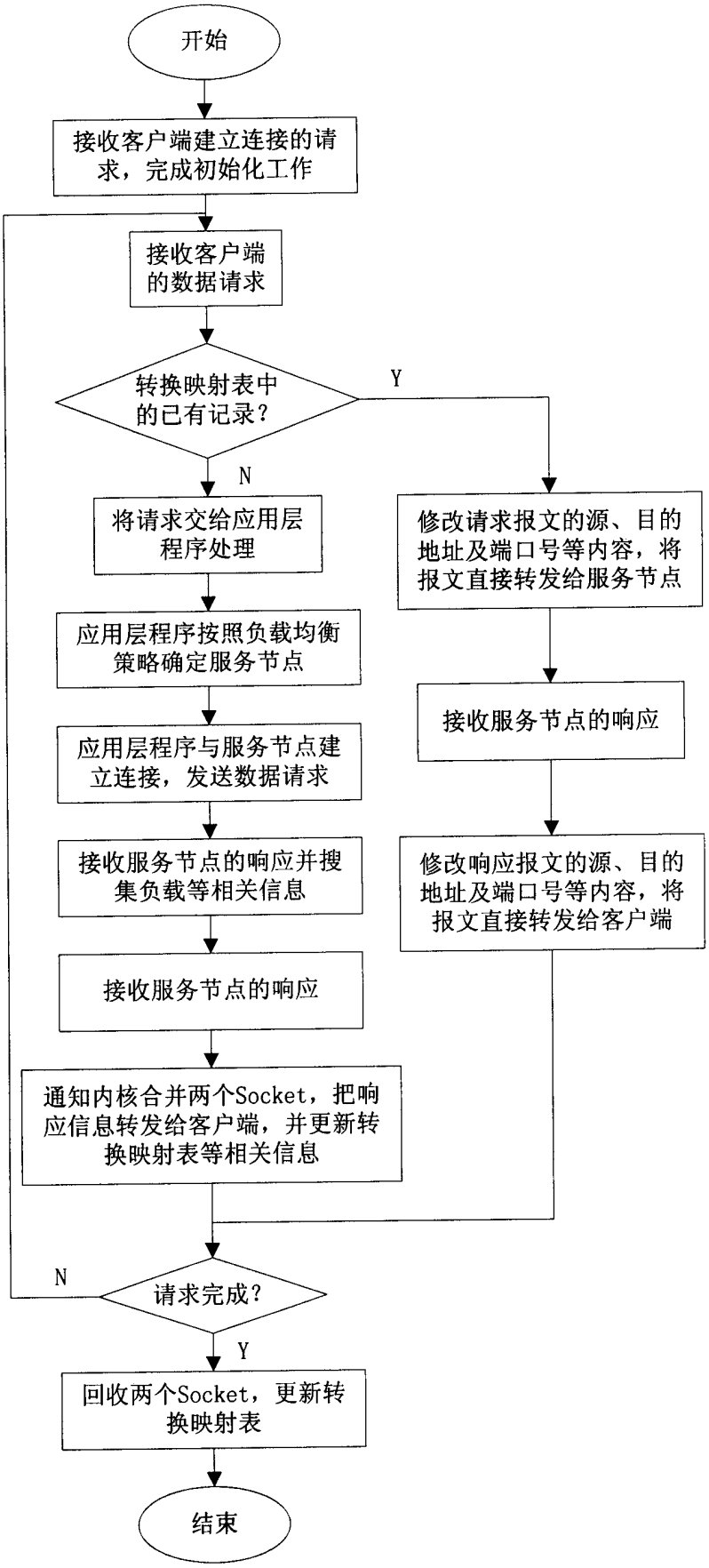

开始
接收客户端建立连接的请求，完成初始化工作
接收客户端的数据请求
转换映射表中的已有记录？
Y
N
将请求交给应用层程序处理
应用层程序按照负载均衡策略确定服务节点
应用层程序与服务节点建立连接，发送数据请求
接收服务节点的响应并搜集负载等相关信息
接收服务节点的响应
通知内核合并两个Socket，把响应信息转发给客户端，并更新转换映射表等相关信息
修改请求报文的源、目的地址及端口号等内容，将报文直接转发给服务节点
接收服务节点的响应
修改响应报文的源、目的地址及端口号等内容，将报文直接转发给客户端
请求完成？
N
Y
回收两个Socket，更新转换映射表
结束

图 3

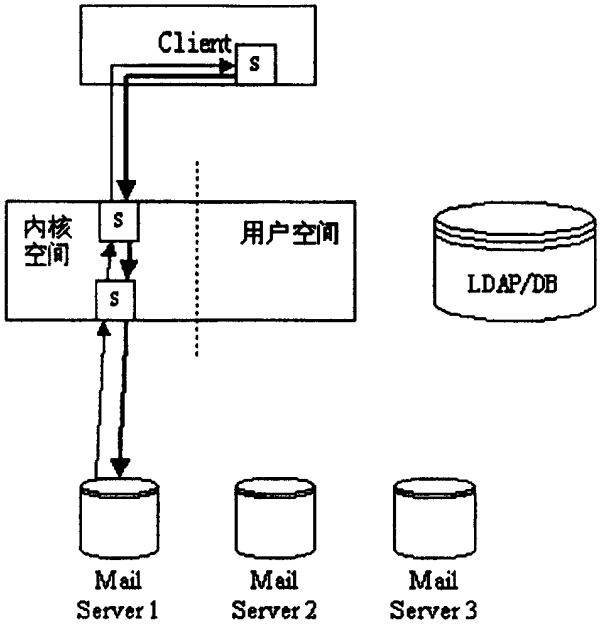

Client
S
内核
空间
S
S
用户空间
LDAP/DB
Mail
Server 1
Mail
Server 2
Mail
Server 3

图 4