



(12) 发明专利申请

(10) 申请公布号 CN 103685331 A

(43) 申请公布日 2014. 03. 26

(21) 申请号 201210318759. 5

(22) 申请日 2012. 08. 31

(71) 申请人 腾讯科技(深圳)有限公司

地址 518000 广东省深圳市福田区赛格科技园 2 栋东 403 室

(72) 发明人 朱念洋

(74) 专利代理机构 北京三高永信知识产权代理有限公司 11138

代理人 鞠永善

(51) Int. Cl.

H04L 29/08(2006. 01)

H04L 12/701(2013. 01)

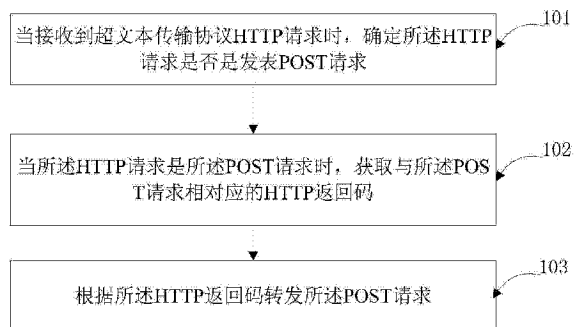
权利要求书2页 说明书10页 附图2页

(54) 发明名称

信息处理的方法及服务器

(57) 摘要

本发明公开了一种信息处理的方法及服务器,属于计算机网络技术领域。所述方法包括:当接收到超文本传输协议 HTTP 请求时,确定所述 HTTP 请求是否是发表 POST 请求;当所述 HTTP 请求是所述 POST 请求时,获取与所述 POST 请求相对应的 HTTP 返回码;根据所述 HTTP 返回码转发所述 POST 请求。实现对 POST 请求的路由转发。



1. 一种信息处理的方法,其特征在于,所述方法包括:
当接收到超文本传输协议 HTTP 请求时,确定所述 HTTP 请求是否是发表 POST 请求;
当所述 HTTP 请求是所述 POST 请求时,获取与所述 POST 请求相对应的 HTTP 返回码;
根据所述 HTTP 返回码转发所述 POST 请求。
2. 根据权利要求 1 所述的方法,其特征在于,所述获取与所述 POST 请求相对应的 HTTP 返回码,包括:
获取所述 POST 请求所携带的 POST 参数;
根据所述 POST 参数获取与所述 POST 请求相对应的 HTTP 返回码。
3. 根据权利要求 2 所述的方法,其特征在于,所述根据所述 POST 参数获取与所述 POST 请求相对应的 HTTP 返回码包括:
根据所述 POST 参数查询本地存储的路由区列表,确定所述 POST 参数是否在所述路由区列表内;
当所述 POST 参数在所述路由区列表内时,则确定与所述 POST 参数相对应的 HTTP 返回码为第一预定值;
当所述 POST 参数不在所述路由区列表内时,则确定与所述 POST 参数对应的 HTTP 返回码为第二预定值。
4. 根据权利要求 1 至 3 任一项所述的方法,其特征在于,所述根据所述 HTTP 返回码转发所述 POST 请求包括:
将所述 POST 请求转发给所述 HTTP 返回码对应的服务器。
5. 根据权利要求 1 所述的方法,其特征在于,所述 HTTP 返回码为非标准的 HTTP 协议返回码。
6. 一种服务器,其特征在于,所述服务器包括:确定模块、获取模块及转发模块;
所述确定模块,用于当接收到超文本传输协议 HTTP 请求时,确定所述 HTTP 请求是否是发表 POST 请求;
所述获取模块,用于当所述 HTTP 请求是所述 POST 请求时,获取与所述 POST 请求相对应的 HTTP 返回码;
所述转发模块,用于根据所述 HTTP 返回码转发所述 POST 请求。
7. 根据权利要求 6 所述的服务器,其特征在于,所述获取模块包括:
第一获取单元,用于获取所述 POST 请求所携带的 POST 参数;
第二获取单元,用于根据所述 POST 参数获取与所述 POST 请求相对应的 HTTP 返回码。
8. 根据权利要求 7 所述的服务器,其特征在于,所述第二获取单元包括:
查找子单元,用于根据所述 POST 参数查询本地存储的路由区列表,确定所述 POST 参数是否在所述路由区列表内;
第一确定子单元,用于当所述 POST 参数在所述路由区列表内时,则确定与所述 POST 参数相对应的 HTTP 返回码为第一预定值;
第二确定子单元,用于当所述 POST 参数不在所述路由区列表内时,则确定与所述 POST 参数对应的 HTTP 返回码为第二预定值。
9. 根据权利要求 6 至 8 任一项所述的服务器,其特征在于,所述转发模块,具体用于将所述 POST 请求转发给所述 HTTP 返回码对应的服务器。

10. 根据权利要求 6 所述的服务器,其特征在于,所述 HTTP 返回码为非标准的 HTTP 协议返回码。

信息处理的方法及服务器

技术领域

[0001] 本发明涉及计算机网络技术领域,特别涉及一种信息处理的方法及服务器。

背景技术

[0002] HTTP (HyperText Transfer Protocol,超文本传输协议)是面向事物的应用层协议,基于该 HTTP 协议,用户端与网络服务器间可以可靠的交换诸如文本、声音或者图像等文件。具体地,用户端可以通过 HTTP 协议中的 GET (获取)请求从网络服务器中获取数据,通过 HTTP 协议中的 POST (发表)请求向网络服务器传送数据。

[0003] Nginx 服务器是一种轻量级的 Web (网页)代理服务器,具有较强的并发处理能力,且可以提供路由功能。现有技术中,Nginx 服务器在接收到 GET 请求后,可以根据 GET 请求中的参数将接收到的 GET 请求转发到相应的网络服务器,以从所述网络服务器中获取数据。但现有技术中,Nginx 服务器并未实现对 POST 请求的路由转发功能。

发明内容

[0004] 为了解决现有技术的问题,本发明实施例提供了一种信息处理的方法及服务器。所述技术方案如下:

[0005] 一方面,提供了一种信息处理的方法,所述方法包括:

[0006] 当接收到超文本传输协议 HTTP 请求时,确定所述 HTTP 请求是否是发表 POST 请求;

[0007] 当所述 HTTP 请求是所述 POST 请求时,获取与所述 POST 请求相对应的 HTTP 返回码;

[0008] 根据所述 HTTP 返回码转发所述 POST 请求。

[0009] 所述获取与所述 POST 请求相对应的 HTTP 返回码,包括:

[0010] 获取所述 POST 请求所携带的 POST 参数;

[0011] 根据所述 POST 参数获取与所述 POST 请求相对应的 HTTP 返回码。

[0012] 所述根据所述 POST 参数获取与所述 POST 请求相对应的 HTTP 返回码包括:

[0013] 根据所述 POST 参数查询本地存储的路由区列表,确定所述 POST 参数是否在所述路由区列表内;

[0014] 当所述 POST 参数在所述路由区列表内时,则确定与所述 POST 参数相对应的 HTTP 返回码为第一预定值;

[0015] 当所述 POST 参数不在所述路由区列表内时,则确定与所述 POST 参数对应的 HTTP 返回码为第二预定值。

[0016] 所述根据所述 HTTP 返回码转发所述 POST 请求包括:

[0017] 将所述 POST 请求转发给所述 HTTP 返回码对应的服务器。

[0018] 所述 HTTP 返回码为非标准的 HTTP 协议返回码。

[0019] 另一方面,提供了一种服务器,所述服务器包括:确定模块、获取模块及转发模

块；

[0020] 所述确定模块,用于当接收到超文本传输协议 HTTP 请求时,确定所述 HTTP 请求是否是发表 POST 请求；

[0021] 所述获取模块,用于当所述 HTTP 请求是所述 POST 请求时,获取与所述 POST 请求相对应的 HTTP 返回码；

[0022] 所述转发模块,用于根据所述 HTTP 返回码转发所述 POST 请求。

[0023] 所述获取模块包括：

[0024] 第一获取单元,用于获取所述 POST 请求所携带的 POST 参数；

[0025] 第二获取单元,用于根据所述 POST 参数获取与所述 POST 请求相对应的 HTTP 返回码。

[0026] 所述第二获取单元包括：

[0027] 查找子单元,用于根据所述 POST 参数查询本地存储的路由区列表,确定所述 POST 参数是否在所述路由区列表内；

[0028] 第一确定子单元,用于当所述 POST 参数在所述路由区列表内时,则确定与所述 POST 参数相对应的 HTTP 返回码为第一预定值；

[0029] 第二确定子单元,用于当所述 POST 参数不在所述路由区列表内时,则确定与所述 POST 参数对应的 HTTP 返回码为第二预定值。

[0030] 所述转发模块,具体用于将所述 POST 请求转发给所述 HTTP 返回码对应的服务器。

[0031] 所述 HTTP 返回码为非标准的 HTTP 协议返回码。

[0032] 本发明实施例提供的技术方案带来的有益效果是：当确定所接收到的 HTTP 请求是所述 POST 请求时,获取与所述 POST 请求相对应的 HTTP 返回码,根据所述 HTTP 返回码转发所述 POST 请求,从而实现了对 POST 请求的路由转发。

附图说明

[0033] 为了更清楚地说明本发明实施例中的技术方案,下面将对实施例描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图仅仅是本发明的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据这些附图获得其他的附图。

[0034] 图 1 是本发明实施例一提供的一种信息处理的方法流程图；

[0035] 图 2 是本发明实施例二提供的一种信息处理的方法流程图；

[0036] 图 3 是本发明实施例三提供的一种服务器的结构示意图；

[0037] 图 4 是本发明实施例三提供的一种服务器的结构示意图。

具体实施方式

[0038] 为使本发明的目的、技术方案和优点更加清楚,下面将结合附图对本发明实施方式作进一步地详细描述。

[0039] 实施例一

[0040] 参见图 1,一种信息处理的方法,所述方法包括：

[0041] 步骤 101:当接收到超文本传输协议 HTTP 请求时,确定所述 HTTP 请求是否是发表

POST 请求；

[0042] 步骤 102:当所述 HTTP 请求是所述 POST 请求时,获取与所述 POST 请求相对应的 HTTP 返回码；

[0043] 步骤 103:根据所述 HTTP 返回码转发所述 POST 请求。

[0044] 本发明实施例中,当确定所接收到的 HTTP 请求是所述 POST 请求时,获取与所述 POST 请求相对应的 HTTP 返回码,根据所述 HTTP 返回码转发所述 POST 请求,从而使不能实现对 POST 请求进行路由转发的服务器,如 Nginx 服务器可以采用该上述方案在接收到 POST 请求后,实现对 POST 请求的路由转发。

[0045] 实施例二

[0046] 参见图 2,本发明实施例提供了一种信息处理的方法,包括如下操作：

[0047] 步骤 201:用户端向 Nginx 服务器发送 HTTP 请求；

[0048] 步骤 202:当 Nginx 服务器接收到 HTTP 请求时,确定所接收到的 HTTP 请求是否是 POST (发表) 请求；

[0049] 当所接收到的 HTTP 请求是 POST 请求时,执行步骤 203；

[0050] 当所接收到的 HTTP 请求不是 POST 请求时,采用现有的方式来处理所接收到的 HTTP 请求；

[0051] 本发明实施例中,HTTP 请求的格式如下：

[0052] <request line> // 请求行

[0053] <headers> // 首部

[0054] <blan line> // 空行

[0055] [<request body>] // 主体

[0056] 在 HTTP 请求中,第一行是请求行 <request line>,用来说明请求资源的 method (方法) (如是 GET 方法还是 POST 方法等)、要访问资源对应的 URI (Uniform Resource Identifier,通用资源标识符) 及使用的 HTTP 协议的版本,紧接着是一个首部 <headers>,用来说明服务器要使用的附加信息(如请求应用的标识 appid),在首部之后是一个空行 <blan line>,再之后就可以添加任意的其他数据(称之为 (body))。

[0057] 基于以上 HTTP 请求格式的描述可以知道,Nginx 服务器在接收到 HTTP 请求后,可以由该 HTTP 请求中请求行内的 method 项来获取该 HTTP 请求是否是 POST 请求。

[0058] 步骤 203:Nginx 服务器获取与所述 POST 请求相对应的 HTTP 返回码；

[0059] 具体地,Nginx 服务器获取所述 POST 请求所携带的 POST 参数；

[0060] Nginx 服务器根据所述 POST 参数获取与所述 POST 请求相对应的 HTTP 返回码。

[0061] 其中,Nginx 服务器根据所述 POST 参数获取与所述 POST 请求相对应的 HTTP 返回码包括：

[0062] Nginx 服务器根据所述 POST 参数查询本地存储的路由区列表,确定所述 POST 参数是否在所述路由区列表内；

[0063] 当所述 POST 参数在所述路由区列表内时,则确定与所述 POST 参数相对应的 HTTP 返回码为第一预定值；

[0064] 当所述 POST 参数不在所述路由区列表内时,则确定与所述 POST 参数对应的 HTTP 返回码为第二预定值。

[0065] 本发明实施例所述的 HTTP 返回码指的是非标准的 HTTP 协议的返回码,所述 POST 参数包括 URI 及 appid。

[0066] 步骤 204 :Nginx 服务器根据所述 HTTP 返回码转发所述 POST 请求 ;

[0067] 具体地,Nginx 服务器将所述 POST 请求转发给所述 HTTP 返回码对应的服务器。

[0068] 具体实现时,Nginx 服务器根据所述 HTTP 返回码查询 HTTP 返回码与服务器间对应关系,得到所述 HTTP 返回码所对应的服务器,之后,将所述 POST 请求转发给查找到的服务器。例如,当所述 HTTP 返回码是第一预定值,如 433 时,查找 HTTP 返回码与服务器间对应关系,得到该 HTTP 返回码 433 对应的服务器为 6788,则将所述 POST 请求转发给服务器 6788 ;当所述 HTTP 返回码是第二预定值,如 434 时,查找 HTTP 返回码与服务器间对应关系,得到该 HTTP 返回码 434 对应的服务器为 6789,则将所述 POST 请求转发给服务器 6789。

[0069] 本发明实施例中,当 Nginx 服务器确定所接收到的 HTTP 请求是所述 POST 请求时,获取与所述 POST 请求对应的 HTTP 返回码,并根据所述 HTTP 返回码转发所述 POST 请求,从而实现对 POST 请求的路由转发。

[0070] 现结合具体的代码程序来详细说明上述技术方案的实现过程,描述如下 :

[0071] 在 Nginx 服务器默认配置支持下,可以根据 POST 请求对应的 HTTP 返回码,将接收到的 POST 请求路由到 HTTP 返回码对应的服务器,一个示例的配置实现如下 :

[0072]

```
server {  
    listen      8080;    //监听 8080 端口  
    server_name localhost;  
    location / {  
        proxy_pass http://localhost:8888; //将监听到的 POST 请求路由到默认的服务器  
8888  
        error_page 433 = @433;    //默认的服务器 8888 返回非 HTTP 协议的返回码 433  
        error_page 434 = @434;    //默认的服务器 8888 返回非 HTTP 协议的返回码 434  
    }  
    location @433 {  
        proxy_pass http://localhost:6788; //当返回码为 433 时，可以将接收到的 POST 请  
求路由到服务器 6788  
    }  
    location @434 {  
        proxy_pass http://localhost:6789; //当返回码为 434 时，可以将接收到的 POST 请  
求路由到服务器 6789  
    }  
    error_page 500 502 503 504 /50x.html;  
    location = /50x.html {  
[0073]                root    html;  
                    }  
    }  
}
```

[0074] 上述的配置实现过程，采用了 433 和 434 这两个非标准的 HTTP 协议的返回码，当然在具体配置时，还可以使用其他的非 HTTP 协议的返回码，本发明实施例对此并不做具体限定；当接收到 POST 请求后，接收到的 POST 请求默认进入服务器 8888，其地址具体为 `http://localhost:8888`，然后根据服务器 8888 返回的 HTTP 返回码是 433 还是 434，将接收到的 POST 请求路由到 433 对应的服务器 6788，其地址具体是 `http://localhost:6788` 或者路由到 434 对应的服务器 6789，其地址具体是 `http://localhost:6789`。

[0075] 那么，为了实现 Nginx 服务器可以根据 POST 请求对应的 HTTP 返回码，将 POST 请求路由到不同的服务器，可以在 Nginx 的自定义模块中，实现根据 POST 请求中的 POST 参数，获取与 POST 请求对应的 HTTP 返回码，进而将接收到的 POST 请求路由到 HTTP 返回码对应的服务器中。其中，Nginx 自定义模块代码如下：

```
[0076] void ngx_http_foo_post_handler(ngx_http_request_t*r) {
```


[0077] // 回调函数, POST 请求全部接收后从这里进入, 读取 POST 请求内的 POST 参数, 且可以产生该 POST 请求对应的响应

[0078]

```
ngx_http_request_body_t* rb = r->request_body;
char* body = NULL;
int body_size = 0;
if (rb && rb->buf)
{
    body = (char*)rb->buf->pos;
    body_size = rb->buf->last - rb->buf->pos;
}
int result = get_route_id(r->connection->log,
                        (int)r->method,
                        (char*)r->uri.data,
                        (char*)r->args.data,
                        body,
                        body_size
```

[0079]

); //根据 POST 请求中的 POST 参数, 获取与该 POST 请求对应的 HTTP 返回码

```

    if (result < 0)
    {
        ngx_log_error(NGX_LOG_ERR, r->connection->log, 0, "get_route_id fail,
result:%d", result);
        result = DFT_ROUTE_ID;
    }
    ngx_http_finalize_request(r, result); //输出 POST 请求对应的 HTTP 返回码
}
static ngx_int_t ngx_http_req_route_handler(ngx_http_request_t *r)
{
    ngx_http_read_client_request_body(r, ngx_http_foo_post_handler);
    return NGX_DONE; // 主 handler 结束
}

```

[0080] 这里, 为了实现根据 POST 请求中的 POST 参数, 获取与 POST 请求对应的 HTTP 返回码, 注册一回调函数 ngx_http_foo_post_handler, 当 POST 请求被全部接收完成时, 该回调函数 ngx_http_foo_post_handle 将会被调用, 对所接收到的 POST 请求进行处理, 处理过程中, 通过调用函数 get_route_id 来获取 POST 请求对应的 HTTP 返回码, 之后, 通过函数 ngx_http_fnalize_request(r, result) 向 Nginx 服务器输出 POST 请求对应的 HTTP 返回码。

[0081] 作为一个优化的处理, 还可以通过在 Nginx 自定义模块中添加 lua 脚本, 使得在获取 POST 请求对应的 HTTP 返回码时, 该 Nginx 自定义模块可以调用所添加的 lua 脚本来实现, 这样可以提高配置的灵活性。其中, lua 脚本的代码示例如下:

[0082]

```

function get_route_id(method, uri, args, body) //获取 POST 请求对应的 HTTP 返回码
    loc, pf, appid = get_need_vals(method, uri, args, body) //获取 POST 请求中的 POST 参
数
    if loc == nil or pf == nil or appid == nil then //
        return OUT_CODE
    end

```

//到这里, 得到所有的 POST 参数 loc、pf 及 appid

[0083]

```

    print (loc, pf, appid)

```

[0084] 在根据 POST 请求中的 POST 参数获取 POST 请求对应的 HTTP 返回码之前, 还预先

配置路由区列表,该路由区列表给出了走该路由区的 POST 参数。

[0085] 例如,走合作区的 pf:locs 列表,该合作区可以包括多个区域,如上海、厦门和南京等地区。

[0086]

```

IN_PARAMS = {
    qzone = {
        '/v3/user/get_info',
        '/req_route',
    },
}
走合作区的 appid 列表
IN_APPIDS = {
    600,
}

```

[0087] 本发明实施例中,合作区对应的 HTTP 返回码为 IN_CODE=434,深圳区对应的 HTTP 返回码为 OUT_CODE=433;

[0088] 之后,就可以根据获取的 POST 请求来查找所配置的路由区列表,获取相应的 HTTP 返回码。具体如下:

[0089]

```

-- 找是否在对应的 url, loc 中
if not is_match_pf_and_loc(pf, loc) then
    return OUT_CODE
end //POST 参数不在路由区列表内时,返回 OUT_CODE,也即返回返回码 433;
-- 找是否在对应的 appid 中
if not is_match_appid(appid) then
    return OUT_CODE
end //POST 参数不在路由区列表时,返回 OUT_CODE,也即返回返回码 433;
return IN_CODE //POST 参数在路由区列表时,返回 IN_CODE,也即返回返回
码 434;

end

```

[0090] 结合了 lua 解析器之后,通过获取 POST 请求对应的 HTTP 返回码就可以修改 lua 脚本而实现,不需要通过重新修改和编译 Nginx 自定义模块代码来获取 POST 请求对应的 HTTP 返回码,可以提高配置的灵活性。

[0091] 基于以上的描述可以得出 Nginx 实现根据 POST 请求对应的 HTTP 返回码,转发

POST 请求到不同服务器的最终配置为：

[0092]

```

server {
    listen      8080; 监听 8080 端口
    server_name localhost;
    location /req_route {
        req_route;//获取 POST 请求对应的 HTTP 返回码
        error_page 433 = @433; 返回非 HTTP 协议的返回码 433
        error_page 434 = @434; 返回非 HTTP 协议的返回码 434
    }
    location @433 {
        proxy_pass http://localhost:6788; //返回码为 433 时，可以将接收到的 POST 请求
路由到服务器 6788
    }
    location @434 {
        proxy_pass http://localhost:6789; 返回码为 434 时，可以将接收到的 POST 请求
路由到服务器 6789
    }
    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
        root    html;
    }
}

```

[0093] 实施例三

[0094] 参见图 3, 本发明实施例提供了一种服务器, 该服务器具体可以与上述方法实施例所述的 Nginx 服务器相同, 包括: 确定模块 301、获取模块 302 及转发模块 303;

[0095] 确定模块 301, 用于当接收到超文本传输协议 HTTP 请求时, 确定所述 HTTP 请求是否是发表 POST 请求;

[0096] 获取模块 302, 用于当所述 HTTP 请求是所述 POST 请求时, 获取与所述 POST 请求相对应的 HTTP 返回码;

[0097] 转发模块 303, 用于根据所述 HTTP 返回码转发所述 POST 请求。

[0098] 参见图 4, 本发明实施例如图 3 所示的获取模块 302 包括:

[0099] 第一获取单元 3021, 用于获取所述 POST 请求所携带的 POST 参数;

[0100] 第二获取单元 3022, 用于根据所述 POST 参数获取与所述 POST 请求相对应的 HTTP

返回码。

[0101] 本发明实施例中,如图 4 所示的第二获取单元 3022 包括:

[0102] 查找子单元,用于根据所述 POST 参数查询本地存储的路由区列表,确定所述 POST 参数是否在所述路由区列表内;

[0103] 第一确定子单元,用于当所述 POST 参数在所述路由区列表内时,则确定与所述 POST 参数相对应的 HTTP 返回码为第一预定值;

[0104] 第二确定子单元,用于当所述 POST 参数不在所述路由区列表内时,则确定与所述 POST 参数对应的 HTTP 返回码为第二预定值。

[0105] 本发明实施例中,转发模块 303,具体用于将所述 POST 请求转发给所述 HTTP 返回码对应的服务器。

[0106] 本发明实施例中,上述的 HTTP 返回码为非标准的 HTTP 协议返回码。

[0107] 本发明实施例提供的技术方案带来的有益效果是:当确定所接收到的 HTTP 请求是所述 POST 请求时,获取与所述 POST 请求相对应的 HTTP 返回码,根据所述 HTTP 返回码转发所述 POST 请求,从而实现了对 POST 请求的路由转发。

[0108] 需要说明的是:上述实施例提供的服务器在进行信息处理时,仅以上述各功能模块的划分进行举例说明,实际应用中,可以根据需要而将上述功能分配由不同的功能模块完成,即将服务器的内部结构划分成不同的功能模块,以完成以上描述的全部或者部分功能。另外,上述实施例提供的服务器与信息处理的方法实施例属于同一构思,其具体实现过程详见方法实施例,这里不再赘述。

[0109] 上述本发明实施例序号仅仅为了描述,不代表实施例的优劣。

[0110] 本领域普通技术人员可以理解实现上述实施例的全部或部分步骤可以通过硬件来完成,也可以通过程序来指令相关的硬件完成,所述的程序可以存储于一种计算机可读存储介质中,上述提到的存储介质可以是只读存储器,磁盘或光盘等。

[0111] 以上所述仅为本发明的较佳实施例,并不用以限制本发明,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。

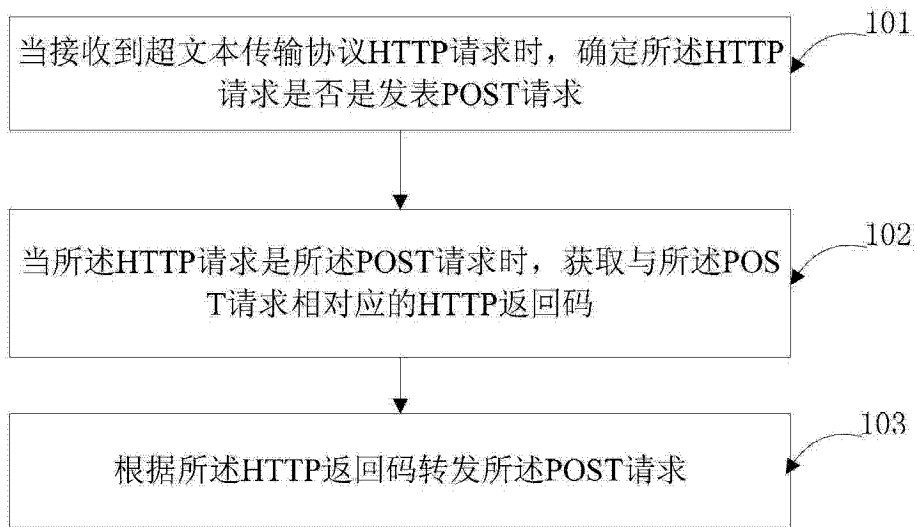


图 1

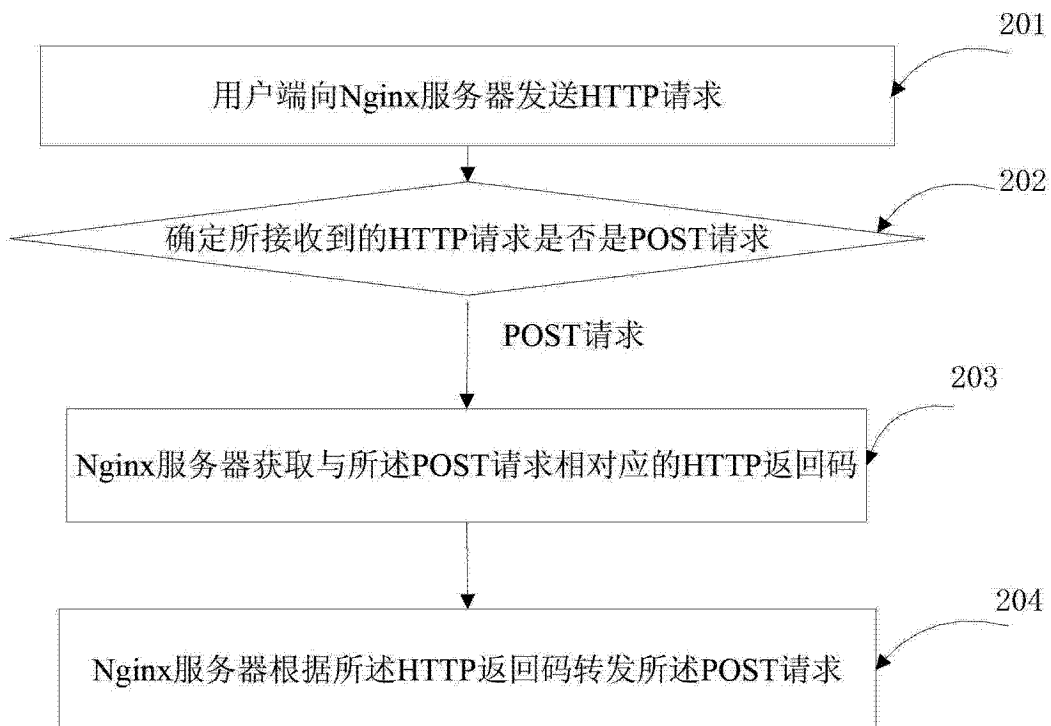


图 2

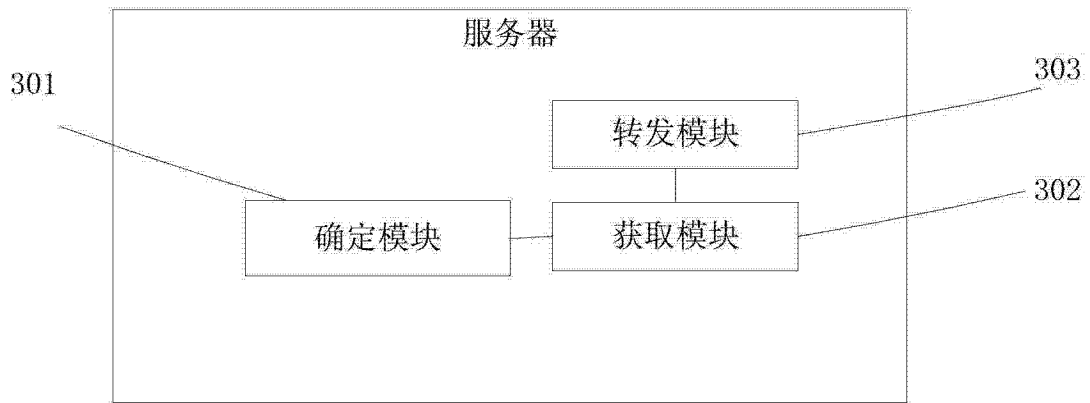


图 3

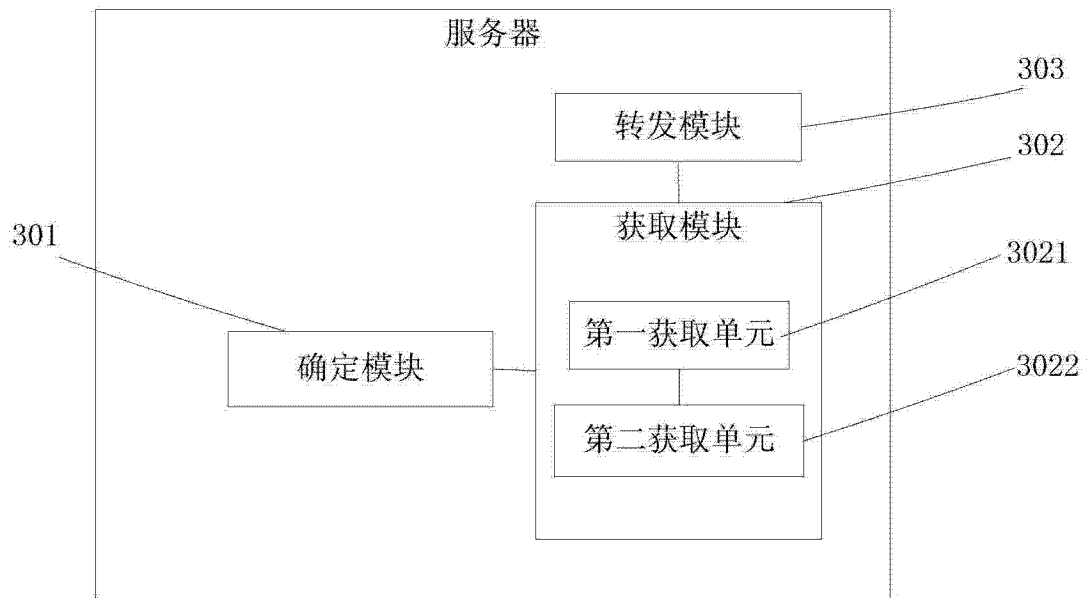


图 4