



(12) 发明专利申请

(10) 申请公布号 CN 102982033 A

(43) 申请公布日 2013. 03. 20

(21) 申请号 201110260094. 2

(22) 申请日 2011. 09. 05

(71) 申请人 深圳市快播科技有限公司

地址 518057 广东省深圳市南山区高新南一道 009 号中国科技开发院中科研发园三号楼 22 层 A

(72) 发明人 曾毅 向灿 伍正勇 钟智将

(74) 专利代理机构 北京三友知识产权代理有限公司 11127

代理人 任默闻

(51) Int. Cl.

G06F 17/30 (2006. 01)

H04L 29/08 (2006. 01)

H04L 29/06 (2006. 01)

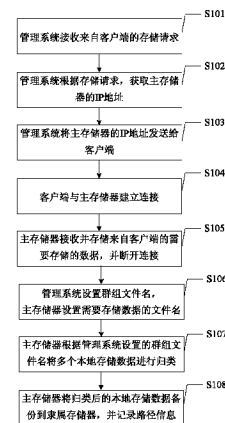
权利要求书 2 页 说明书 9 页 附图 8 页

(54) 发明名称

小文件的存储方法及系统

(57) 摘要

本发明提供一种小文件的存储方法及系统, 其中, 该方法包括: 管理系统接收来自客户端的存储请求; 管理系统根据存储请求, 获取主存储器的 IP 地址; 管理系统将主存储器的 IP 地址发送给客户端; 客户端与主存储器之间建立连接; 主存储器接收并存储来自客户端的需要存储的数据, 并断开客户端与主存储器建立的连接; 管理系统设置群组文件名, 主存储器设置需要存储数据的文件名; 主存储器根据群组文件名将多个本地存储数据进行归类处理; 主存储器将归类处理后的本地存储数据备份到隶属存储器, 并记录存储路径信息。通过本发明, 可以提高 IO 性能。



1. 一种小文件的存储方法,其特征在于,所述的方法包括:
  - 管理系统接收来自客户端的存储请求;
  - 所述管理系统根据所述的存储请求,获取主存储器的 IP 地址;
  - 所述管理系统将所述主存储器的 IP 地址发送给所述客户端;
  - 所述客户端与所述主存储器之间建立连接;
  - 所述主存储器接收并存储来自所述客户端的需要存储的数据,并断开所述客户端与所述主存储器建立的连接;
  - 所述管理系统设置群组文件名,所述主存储器设置所述需要存储数据的文件名;
  - 所述主存储器根据所述群组文件名将多个本地存储数据进行归类处理;
  - 所述主存储器将归类处理后的本地存储数据备份到隶属存储器,并记录存储路径信息。
2. 根据权利要求 1 所述的方法,其特征在于,在获取主存储器的 IP 地址之前,所述的方法还包括:
  - 所述管理系统在存储器组中选择一个存储器为主存储器,其余存储器为隶属存储器,其中,所述的存储器组至少包括两个存储器;
  - 所述管理系统将选择的主存储器的信息发送给选择的隶属存储器。
3. 根据权利要求 2 所述的方法,其特征在于,在所述管理系统将选择的主存储器的信息发送给选择的隶属存储器之后,所述的方法还包括:
  - 当所述的主存储器停机时,所述管理系统在所述存储器组中重新选择新的主存储器。
4. 根据权利要求 1 所述的方法,其特征在于,在所述主存储器将归类处理后的本地存储数据备份到隶属存储器并记录存储路径信息之后,所述的方法还包括:
  - 所述主存储器接收所述的客户端对存储数据的编辑操作;
  - 所述主存储器对编辑后的数据进行保存并备份到相应的隶属存储器。
5. 一种小文件的存储系统,其特征在于,所述的系统包括:管理系统、客户端、以及包括主存储器和隶属存储器的存储器组,其中,
  - 所述的管理系统包括:
    - 存储请求接收单元,用于接收来自客户端的存储请求;
    - IP 地址获取单元,用于根据所述的存储请求获取主存储器的 IP 地址;
    - IP 地址发送单元,用于将所述主存储器的 IP 地址发送给所述客户端;
    - 群组文件名设置单元,用于设置所述主存储器存储数据的群组文件名;
  - 所述主存储器包括:
    - 数据接收单元,用于接收并存储来自所述客户端的需要存储的数据;
    - 通信链路管理单元,用于建立或断开所述客户端与所述主存储器之间的通信链路;
    - 文件名设置单元,用于设置所述需要存储数据的文件名;
    - 数据归类处理单元,用于根据所述管理系统设置的群组文件名将多个本地存储数据进行归类处理;
    - 数据备份单元,用于将归类处理后的本地存储数据备份到隶属存储器,并记录存储路径信息。
6. 根据权利要求 5 所述的系统,其特征在于,所述的管理系统还包括:

隶属关系选择单元,用于在存储器组中选择一个存储器为主存储器,其余存储器为隶属存储器;

隶属信息发送单元,用于将选择的主存储器的信息发送给选择的隶属存储器。

7. 根据权利要求 6 所述的系统,其特征在于,所述的隶属关系选择单元还用于:当所述的主存储器停机时,在所述存储器组中重新选择新的主存储器。

8. 根据权利要求 5 所述的系统,其特征在于,所述主存储器还包括:

编辑操作接收单元,用于接收所述的客户端对存储数据的编辑操作;

所述的数据备份单元还用于对编辑后的数据进行保存并备份到相应的隶属存储器。

## 小文件的存储方法及系统

### 技术领域

[0001] 本发明涉及通信领域,具体地,涉及一种小文件的存储方法及系统。

### 背景技术

[0002] 目前,文件存储系统已趋于成熟。申请号 201010184752.X 公开了一种文件存储方法,该方法通过用户在网页页面上提交文件、并上传至中转服务器,将文件从中转服务器通过远程服务存储到存储服务器。该存储服务器没有独立的单做用作全局管理调度的管理服务器,难以实现全面的全局调度。而且,客户端(Client)分别与主存储器、隶属存储器均建立了短连接,同时完成存储和备份,在主存储器以及隶属存储器均操作成功时,才反馈给 Client 操作成功消息,这属于同步备份,反馈时延较长。并且,存储器的存储用磁盘没有存储区分,这导致了访问的 IO 性能较差,从而可能导致寻址效率低。

[0003] 也就是说,现有的文件存储系统存在反馈时延较长、访问 IO 性能较差的问题。

### 发明内容

[0004] 本发明实施例的主要目的在于提供一种小文件的存储方法及系统,以解决现有技术中的文件存储系统的反馈时延较长、访问 IO 性能较差的问题。

[0005] 为了实现上述目的,本发明实施例提供一种小文件的存储方法,该方法包括:管理系统接收来自客户端的存储请求;所述管理系统根据所述的存储请求,获取主存储器的 IP 地址;所述管理系统将所述主存储器的 IP 地址发送给所述客户端;所述客户端与所述主存储器之间建立连接;所述主存储器接收并存储来自所述客户端的需要存储的数据,并断开所述客户端与所述主存储器建立的连接;所述管理系统设置群组文件名,所述主存储器设置所述需要存储数据的文件名;所述主存储器根据所述群组文件名将多个本地存储数据进行归类处理;所述主存储器将归类处理后的本地存储数据备份到隶属存储器,并记录存储路径信息。

[0006] 具体地,在获取主存储器的 IP 地址之前,上述的方法还包括:所述管理系统在存储器组中选择一个存储器为主存储器,其余存储器为隶属存储器,其中,所述的存储器组至少包括两个存储器;所述管理系统将选择的主存储器的信息发送给选择的隶属存储器。

[0007] 优选地,在所述管理系统将选择的主存储器的信息发送给选择的隶属存储器之后,所述的方法还包括:当所述的主存储器停机时,所述管理系统在所述存储器组中重新选择新的主存储器。

[0008] 优选地,在所述主存储器将归类处理后的本地存储数据备份到隶属存储器并记录存储路径信息之后,所述的方法还包括:所述主存储器接收所述的客户端对存储数据的编辑操作;所述主存储器对编辑后的数据进行保存并备份到相应的隶属存储器。

[0009] 本发明实施例还提供一种小文件的存储系统,该系统包括:管理系统、客户端、以及包括主存储器和隶属存储器的存储器组,其中,所述的管理系统包括:存储请求接收单元,用于接收来自客户端的存储请求;IP 地址获取单元,用于根据所述的存储请求获取主

存储器的 IP 地址 ;IP 地址发送单元,用于将所述主存储器的 IP 地址发送给所述客户端 ;群组文件名设置单元,用于设置所述主存储器存储数据的群组文件名 ;所述主存储器包括 :数据接收单元,用于接收并存储来自所述客户端的需要存储的数据 ;通信链路管理单元,用于建立或断开所述客户端与所述主存储器之间的通信链路 ;文件名设置单元,用于设置所述需要存储数据的文件名 ;数据归类处理单元,用于根据所述管理系统设置的群组文件名将多个本地存储数据进行归类处理 ;数据备份单元,用于将归类处理后的本地存储数据备份到隶属存储器,并记录存储路径信息。

[0010] 具体地,所述的管理系统还包括 :隶属关系选择单元,用于在存储器组中选择一个存储器为主存储器,其余存储器为隶属存储器 ;隶属信息发送单元,用于将选择的主存储器的信息发送给选择的隶属存储器。

[0011] 优选地,所述的隶属关系选择单元还用于 :当所述的主存储器停机时,在所述存储器组中重新选择新的主存储器。

[0012] 优选地,所述主存储器还包括 :编辑操作接收单元,用于接收所述的客户端对存储数据的编辑操作 ;所述的数据备份单元还用于对编辑后的数据进行保存并备份到相应的隶属存储器。

[0013] 借助于上述技术方案至少之一,通过管理系统将主存储器的 IP 地址发送给客户端,使得客户端与主存储器之间可以直接连接进行数据的存储,且,备份操作也是在主存储器和隶属存储器之间实现,从而可以克服现有技术中的反馈时延较长、访问 IO 性能较差的问题,提高 IO 性能。

## 附图说明

[0014] 为了更清楚地说明本发明实施例或现有技术中的技术方案,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性的前提下,还可以根据这些附图获得其他的附图。

[0015] 图 1 是根据本发明实施例的小文件存储方法的流程图 ;

[0016] 图 2 是根据本发明实施例的管理系统与多个存储器之间的连接关系示意图 ;

[0017] 图 3 是根据本发明实施例的小文件存储系统结构示意图 ;

[0018] 图 4、5 分别是 Master、Slave 执行数据备份时的流程图 ;

[0019] 图 6 是 Block 文件的逻辑结构示意图 ;

[0020] 图 7、8 分别是 Master、Slave 处理同一个 Block 的流程图 ;

[0021] 图 9 是根据本发明实施例的小文件存储系统的结构框图 ;

[0022] 图 10 是根据本发明实施例的管理系统的结构框图 ;

[0023] 图 11 是根据本发明实施例的小文件存储系统的架构示意图 ;

[0024] 图 12 是根据本发明实施例的上传文件数据流程 ;

[0025] 图 13 是根据本发明实施例的查阅文件的流程图。

## 具体实施方式

[0026] 下面将结合本发明实施例中的附图,对本发明实施例中的技术方案进行清楚、完

整地描述,显然,所描述的实施例仅仅是本发明一部分实施例,而不是全部的实施例。基于本发明中的实施例,本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例,都属于本发明保护的范围。

[0027] 本发明实施例提供一种小文件的存储方法和系统。以下结合附图对本发明进行详细说明。

[0028] 实施例一

[0029] 本发明实施例提供一种小文件的存储方法,如图 1 所示,该方法包括:

[0030] 步骤 101,管理系统接收来自客户端的存储请求;

[0031] 步骤 102,管理系统根据存储请求,获取主存储器的 IP 地址;

[0032] 步骤 103,管理系统将主存储器的 IP 地址发送给客户端;

[0033] 步骤 104,客户端与主存储器之间建立连接;

[0034] 步骤 105,主存储器接收并存储来自客户端的需要存储的数据,之后断开客户端与主存储器建立的连接;

[0035] 步骤 106,管理系统设置群组文件名,主存储器设置需要存储数据的文件名;

[0036] 步骤 107,主存储器根据管理系统设置的群组文件名将多个本地存储数据进行归类处理;

[0037] 步骤 108,主存储器将归类处理后的本地存储数据备份到隶属存储器,并记录存储路径信息。

[0038] 由以上描述可知,通过管理系统将主存储器的 IP 地址发送给客户端,使得客户端与主存储器之间可以直接连接进行数据的存储,且,备份操作也是在主存储器和隶属存储器之间实现,无需像现有技术中的同步备份,因此,通过本发明实施例可以克服现有技术中的反馈时延较长、访问 IO 性能较差的问题,提高 IO 性能。

[0039] 在实际操作中,在获取主存储器的 IP 地址之前,管理系统需要在存储器组中选择一个存储器为主存储器,其余存储器为隶属存储器,其中,存储器组至少包括两个存储器;然后管理系统将选择的主存储器的信息发送给选择的隶属存储器。

[0040] 图 2 示出了管理系统(也可以称为管理服务器,NameServer)与多个存储器(DataServer)之间的连接关系,如图 2 所示,以两台 DataServer(简称为 DS)为一存储器组(group),布置多组为一集群,每一组由唯一的 ID 号标识,在同一组内 DataServer 有不同的机器号标识。DataServer 在启动时,向 NameServer 注册,提供 ID 号和机器号,NameServer 选择一台 DataServer 作为主存储器(Master),其余的为隶属存储器(Slave)。

[0041] 在备份过程中,由 NameServer 提供给 Master 相应的 Slave 信息。当 Master 因故障停机,那么由 NameServer 自动选择 Slave 作为 Master 继续提供服务,支持 Master 与 Slave 的自动切换,避免人工干预,减少服务停止时间以及人为出错风险。即,当主存储器停机时,管理系统在存储器组中重新选择新的主存储器。在 Master 和 Slave 自动切换时,为避免数据重复备份,采用以下策略:制定 Master 与 Slave 的通信协议,Master 与 Slave 相互通信实时同步备份的位置,记录相应的位置至本地文件。

[0042] 在具体实施过程中,主存储器还可以接收客户端对存储数据的编辑操作,当客户端完成编辑操作后,主存储器对编辑后的数据进行保存并备份到相应的隶属存储器,以便后续的查阅。该编辑操作可以包括查询、修改等操作。

[0043] 以下给出一个实例。

[0044] 图 3 是小文件存储系统结构示意图,如图 3 所示,系统中的每组 MasterServer/SlaveServer 均需向管理服务器 (NameServer, 简称为 Ns) 注册后才能发挥系统存储的作用。注册后的若干组 MasterServer/SlaveServer, 每个 MasterServer 以及 SlaveServer 均通过心跳通信的方式与管理服务器保持长连接,使得 MasterServer/SlaveServer 均受到管理服务器的监控。

[0045] 当用户需要上传数据时,客户端向管理服务器发起上传请求,管理服务器选择合适主存储器,并将该主存储器的连接信息 IP 和 Port 发送给用客户端,客户端根据连接信息与该主存储器建立短连接。连接建立后,主存储器向管理服务器发出 Block 分配请求,该 Block 即上述的群组文件名,由管理服务器设定。管理服务器的 Block 分配模块响应请求并分配 Block id 给主存储器。同时,管理服务器在关系列表模块中建立 Block id 和 DS 的对应关系。DS 在收到分配的 Block id 后,开始和用户进行上传小文件步骤。在实际操作中,主存储器会根据自身情况选择是否需要向管理服务器发送 Block 分配请求。如果主存储器当前大文件 (Block) 写满了,那么主存储器向管理服务器发送 Block 分配请求,Block 分配请求只是分配可用的 Block id,实际存取的大文件名由存储器决定,管理服务器管理 Block id 号,使得全局唯一,避免重复。

[0046] 主存储器将接收到的大量小文件 (实际数据文件) 合并成为一个大文件 (Block)。每一个 Block 拥有集群内唯一的标识 Block id。Block id 由 NameServer 指定和命名,NameServer 维护 Block 和 DataServer 的对应关系,以及 DataServer 相关信息,包括 DataServer 加入与退出。数据存储在 DataServer,各组 DataServer 之间的数据是相互独立的,同一组内的数据是一样的。

[0047] 每一个小文件在系统中对应于唯一的文件名 (Filename),在一个 Block 内有唯一的 File id。这样通过 Block id 和 File id 就可以定位到实际的文件。上述 Filename 就是由 Block id、File id、文件类型以及应用 ID 号编码生成。

[0048] 在用户 (Client) 上传数据时,只和主存储器建立短连接,不和隶属存储器建立直接连接。在用户和主存储器通讯完成并反馈给用户操作成功消息后即断开。

[0049] 当用户需要进行查询或删除操作时,客户端向管理服务器发起查询或删除请求,管理服务器根据查询或删除的对象 ID,从关系列表模块中找到该对象 ID 对应的 Block id 以及对应的 DataServer。管理服务器将 DataServer 的连接信息 IP 和 Port 发送给用户,用户根据连接信息与管理服务器确定的 DataServer 建立短连接,连接后在具体的 Block 内进行相应的查询或删除操作。

[0050] 主存储器在设定的时间间隔或者设定文件修改操作量条件满足时,在主存储器和隶属存储器之间完成异步备份作业。NameServer 和 DataServer 都有热备份存在,当 NameServer 或 DataServer 不可用,相应的热备就可以充当 NameServer 或 DataServer 使用。

[0051] 上述数据备份采用的是 Master-Slave 架构,Master 对数据更新操作时,如执行完上传或删除操作时,将更新信息串行化写入本地 log 文件。以每一组为单位,在同组内从 Master 异步备份数据至 Slave。Master 根据 log 文件,将数据操作发送至 Slave,Slave 记录当前备份所处位置。

[0052] 备份的数据分为两部分：实际文件数据和信息数据，以下分别对这两部分数据的备份进行描述。

[0053] （一）对实际文件数据的备份

[0054] 制定 Master 与 Slave 的通信协议，Master 与 Slave 相互通信实时同步备份的位置，记录相应的位置至本地文件 log1。同时，Slave 记录作为 Slave 时，同步数据所处的位置至本地文件 log2。当 Slave 切换成 Master 时，新的 Slave 向新的 Master 发送数据同步请求，新的 Master 根据 log2 记录的文件坐标，定位至相应的位置开始同步数据；如果没有发生 Master 与 Slave 切换，Slave 就根据 log1 记录的文件坐标，向 Master 发送数据同步请求，Master 根据 Slave 发送的文件坐标位置，定位到相应的位置同步数据。坐标位置表示为：文件名和偏移量。

[0055] 具体地，Master 开启一个线程，作为同步的 Server 端，Slave 开启一个线程，作为 Client 端。Server 和 Client 工作流程分别如下：

[0056] （1）Client 端：

[0057] 1) Client 向 Server 发送数据同步请求，并告知当前数据所处的坐标位置 (logName, logPos)，如果发送请求失败，则休眠 1 秒钟，继续发送请求，如果成功，则进行至 2)；

[0058] 2) Client 等待接收 Server 发送指令和数据，如果成功进行至 3)；

[0059] 3) 解析指令，进行文件数据写入或删除操作，以及坐标位置更新操作，如果成功进行至 4)，否则返回至 1)；

[0060] 4) 将坐标位置写入本地文件，同时更新 logName 和 logPos，返回至 1)。

[0061] （2）Server 端：

[0062] 1) Server 等待接收 Client 的同步请求，如果成功进行至 2)；

[0063] 2) Server 根据 Client 的坐标位置，定位到相应位置，如果有数据，就发送给 Client，否则进行至 5)，如果成功发送数据则进行至 3)，失败则返回至 1)；

[0064] 3) Server 向 Client 发送数据到相应的坐标位置，成功则跳至 4)，否则返回至 1)；

[0065] 4) 更新坐标位置，返回至 2)；

[0066] 5) 检查是否要跳转至新的 log 文件，如果是，则更新坐标位置信息，并发送该信息给 Client，然后返回至 1)。

[0067] 图 4、5 分别是 Master、Slave 执行数据备份时的流程图，如图 4 所示，Master 的操作流程为：

[0068] 步骤 401，等待接收数据同步请求；

[0069] 步骤 402，根据坐标位置，定位至相应的位置；

[0070] 步骤 403，判断是否有未同步的数据，如果有，则进行步骤 404，否则进行步骤 405；

[0071] 步骤 404，发送数据给 Client，即，发送数据给 Slave，然后进行步骤 406；

[0072] 步骤 405，检查是否需要转至新文件，如果是，更新坐标位置，并发送给 Client；

[0073] 步骤 406，是否发送数据成功，如果是，则进行步骤 407，否则返回至步骤 401；

[0074] 步骤 407，发送新的坐标位置给 Client；

[0075] 步骤 408，是否发送新的坐标位置成功，如果是，则进行步骤 409，否则返回至步骤 401；



- [0076] 步骤 409,更新坐标位置。
- [0077] 如图 5 所示,Slave 的操作流程为;
- [0078] 步骤 501,根据坐标位置,发送数据同步请求;
- [0079] 步骤 502,是否发送数据同步请求成功,如果是,则进行步骤 503,否则进行步骤 504;
- [0080] 步骤 503,等待接收数据和指令;
- [0081] 步骤 504,休眠 1 秒,并返回步骤 501;
- [0082] 步骤 505,是否有数据,如果是,则进行步骤 506,否则返回步骤 501;
- [0083] 步骤 506,接收数据,解析指令,执行相关操作;
- [0084] 步骤 507,更新坐标位置信息,并写入本地文件。

[0085] 由上述可知,Master 根据 Slave 发送的坐标信息,将相对于该位置,做过更新操作的数据全部发送给 Slave,直到没有数据可以发送;而 Slave 一直接收数据,直到数据到达,然后根据最新的坐标位置信息,向 Master 发送同步请求。如果数据同步不成功,那么坐标信息也不会更新,此时 Slave 会要求 Master 再一次发送该数据,直到成功。

#### [0086] (二) 对信息数据的备份

[0087] 信息数据备份针对每一个 Block,在 Master 和 Slave 之间,维持同一个能分配使用的 File id,保持实时的一致性,同时写入该 File id 至对应 Block 文件的头部,每个 Block 文件逻辑结构如图 6 所示,每一 Block 有一个 8 字节的 Block head 用于存放针对该 Block 能使用的下一个 File id,在 Block 中,每一小文件都有一个 32 字节的 file head,存放索引信息。

[0088] 在系统中可以建立一个 Block 索引数据库,用于存放每一个 Block 能分配使用的 File id,在同一个 Block 中 File id 加 1 递增。Block 每分配一个 File id,那么就更新对应 Block 的数据库记录,同时写入 Block 文件头部,并且将该 File id 同步至 Slave,更新 Slave 上对应的数据库记录和 Block 文件头部。如果同步 File id 至 Slave,则保存该记录,等待重传。需要说明的是,在重传的记录中,每一个 Block id 只对应一条 File id 记录,并且该记录可能会被实时更新。

[0089] 图 7、8 分别是 Master、Slave 处理同一个 Block 的流程图,如图 7 所示,Master 的操作流程为:

- [0090] 步骤 701,Block 分配一个 File id;
- [0091] 步骤 702,将下一个可用的 File id 保存至相应的数据库、并写入 Block 头部;
- [0092] 步骤 702,检测是否有需要重传的记录,如果是,则进行步骤 704,否则,进行步骤 706;
- [0093] 步骤 704,是否重传成功,如果否,则进行步骤 705,否则进行步骤 706;
- [0094] 步骤 705,留待下一次重传;
- [0095] 步骤 706,删除重传成功的记录,并同步 File id 至 Slave;
- [0096] 步骤 707,是否同步成功,如果否,则进行步骤 708,否则返回步骤 701;
- [0097] 步骤 708,保存和更新重传记录。
- [0098] 如图 8 所示,Slave 的操作流程为:
- [0099] 步骤 801,接收同步的 Block id 和 file id;

[0100] 步骤 802,将 Block id 和 file id 保存至相应的数据库、写入 Block 头部,并返回步骤 802。

[0101] 实施例二

[0102] 本发明实施例还提供一种小文件的存储系统,如图 9 所示,该系统包括:管理系统 1、客户端 2、以及包括主存储器 31 和隶属存储器 32 的存储器组 3,其中:

[0103] 管理系统 1 包括:

[0104] 存储请求接收单元 11,用于接收来自客户端的存储请求;

[0105] IP 地址获取单元 12,用于根据存储请求获取主存储器的 IP 地址;

[0106] IP 地址发送单元 13,用于将主存储器的 IP 地址发送给客户端;

[0107] 群组文件名设置单元 14,用于设置主存储器存储数据的群组文件名;

[0108] 主存储器 31 包括:

[0109] 数据接收单元 311,用于接收并存储来自客户端的需要存储的数据;

[0110] 文件名设置单元 312,用于根据预定规则设置需要存储数据的文件名;

[0111] 数据归类处理单元 313,用于根据管理系统设置的群组文件名将多个本地存储数据进行归类处理;

[0112] 数据备份单元 314,用于将归类处理后的本地存储数据备份到隶属存储器,并记录存储路径信息;

[0113] 通信链路管理单元 315,用于建立或断开客户端与主存储器之间的通信链路。

[0114] 由以上描述可以看出,通过管理系统将主存储器的 IP 地址发送给客户端,使得客户端与主存储器之间可以直接联系进行数据的存储,且,备份操作也是在主存储器和隶属存储器之间实现,无需像现有技术中的同步备份,因此,通过本发明实施例可以克服现有技术中的反馈时延较长、访问 IO 性能较差的问题,提高 IO 性能。

[0115] 如图 10 所示,管理系统 1 还包括:

[0116] 隶属关系选择单元 15,用于在存储器组中选择一个存储器为主存储器,其余存储器为隶属存储器;

[0117] 隶属信息发送单元 16,用于将选择的主存储器的信息发送给选择的隶属存储器。

[0118] 当主存储器停机时,隶属关系选择单元 14 还用于在存储器组中重新选择新的主存储器。

[0119] 上述主存储器 31 还包括:编辑操作接收单元,用于接收客户端对存储数据的编辑操作。相应地,数据备份单元 314 还用于对编辑后的数据进行保存并备份到相应的隶属存储器。

[0120] 上述各单元具体的执行过程,可以参考上述实施例一中的描述,此处不再赘述。

[0121] 为了更好的理解本发明实施例,以下结合图 11 详细描述本发明实施例。图 11 是该小文件存储系统的架构示意图,如图 11 所示,小文件存储系统(以下简称为 kwfs)是一种分布式文件存储系统,由三大部分组成:kwfs Client, kwfs NameSverver 和 kwfs DataServer。kwfs NameSverver 为中心节点,作为整个小文件存储系统的调度和控制中心,DataServer 可以自由添加和移除,保证良好的可扩展性。同时,对应于若干 DataServer,均设置 SlaveServer,作为 DataServer 的备份。其中,DataServer 和 SlaveServer 均与 NameSverver 建立采用心跳机制的长连接。

[0122] 其中,NameSverver 管理各 DataServer 有关信息,包括系统信息、各 DataServer 服务器加入、退出等、以及管理各 DataServer 服务器 Block id 的创建,删除,负载等。

[0123] DataServer 处理实际数据存储与读写,向 NameServer 报告服务器状态(负载,文件数等);维护 Block id 与文件 File id 关系(索引)数据存储位置、大小等。

[0124] 图 12 是上传文件数据流程,如图 12 所示,当 Client 向 NameSverver 发出请求信息时,NameSverver 查询本地由 DataServer 不断更新的索引目录并将具有被查文件数据的 DataServer 的 IP 地址和 Port 反馈给 Client,Client 再分别与 DataServer 建立数据通信连接,以保证 NameSverver 的正常运作减少传输压力。其中,该不断更新的索引目录(Master,Slave)DataServer 在本地也保存。在完成上传之后,DataServer 将 Block 状态更新到 NameSverver,并向 Client 反馈操作结果,如包括文件名(Filename)。

[0125] 文件名由 Block id、File id、文件类型和应用 id 组成,采用 base64(六位 2 进制表示)编码生成。如果使用 8 字节存储 Block id,4 字节存储 File id,那么 Filename 的长度为 16 个字节。如:假设 Block id 为 123456,File id 为 888 那么 kwfs 文件名为:AAAAAAB4kAAAAAN4。如果需要标识文件类型,那么设定六位二进制组合表示文件类型(可以标识 64 种文件类型),那么 kwfs 文件名长度为 17 个字节。

[0126] 在 NameSverver 上存储 Block id 和 DataServer 的对应关系,在 DataServer 存储 Block id 和 File id 以及文件在 Block 内的偏移量和文件长度的对应关系。

[0127] 图 13 是查阅文件的流程图,如图 13 所示,Client 根据 Filename 解码出 Block id 和 File id(File id 是相对于 Block id 的),然后定位到实际数据。

[0128] 当需要删除文件时,首先是逻辑删除,即在 DataServer 上将相应的 File id 的信息移出至保存删除信息的数据库中,然后是物理删除,在特定的时间点,根据删除信息回收对应文件的磁盘空间,按序移动和合并 Block 中未删除的数据,修改相应 File id 的偏移量。

[0129] 在备份策略上,可采用多写和单写备份方式:多写方式在一定程度上会影响 Server 的响应速度;单写则需要以日志的方式记录数据备份的时间点和状态,保证在备份过程中出现故障时,再次备份时不会丢失数据。在系统实现过程中,可以根据实际情况选择合适的策略。

[0130] 由以上描述可以看出,本发明实施例通过由 NameSverver 统一全局调股管理,扩展时直接增加 Master DataServer/Slave DataServer 组即可,实现对 DataServer 的统一全局管理调度,可扩展性好。该 NameSverver 的设置可完成 Master 与 Slave 自动切换,避免人工干预,也可以避免切换后数据重复备份,减少服务停止时间。且,Client 只和 Master DataServer 建立短连接,直接通信完成并反馈给 Client 操作成功消息后即断开。Master DataServer 在设定的时间间隔或者设定文件修改操作量条件满足时,和 Slave DataServer 之间完成异步备份作业。并且,由于若干 DataServer 的设置,使得整个小文件系统的容错性增强,同时,NameSverver 对 Client 的统一请求管理,和 Client 与 NameSverver 的通信方法,使得系统的稳定性增强。

[0131] 本领域普通技术人员可以理解实现上述实施例方法中的全部或部分步骤可以通过程序来指令相关的硬件来完成,该程序可以存储于一计算机可读取存储介质中,比如 ROM/RAM、磁碟、光盘等。

[0132] 以上所述的具体实施例,对本发明的目的、技术方案和有益效果进行了进一步详细说明,所应理解的是,以上所述仅为本发明的具体实施例而已,并不用于限定本发明的保护范围,凡在本发明的精神和原则之内,所做的任何修改、等同替换、改进等,均应包含在本发明的保护范围之内。

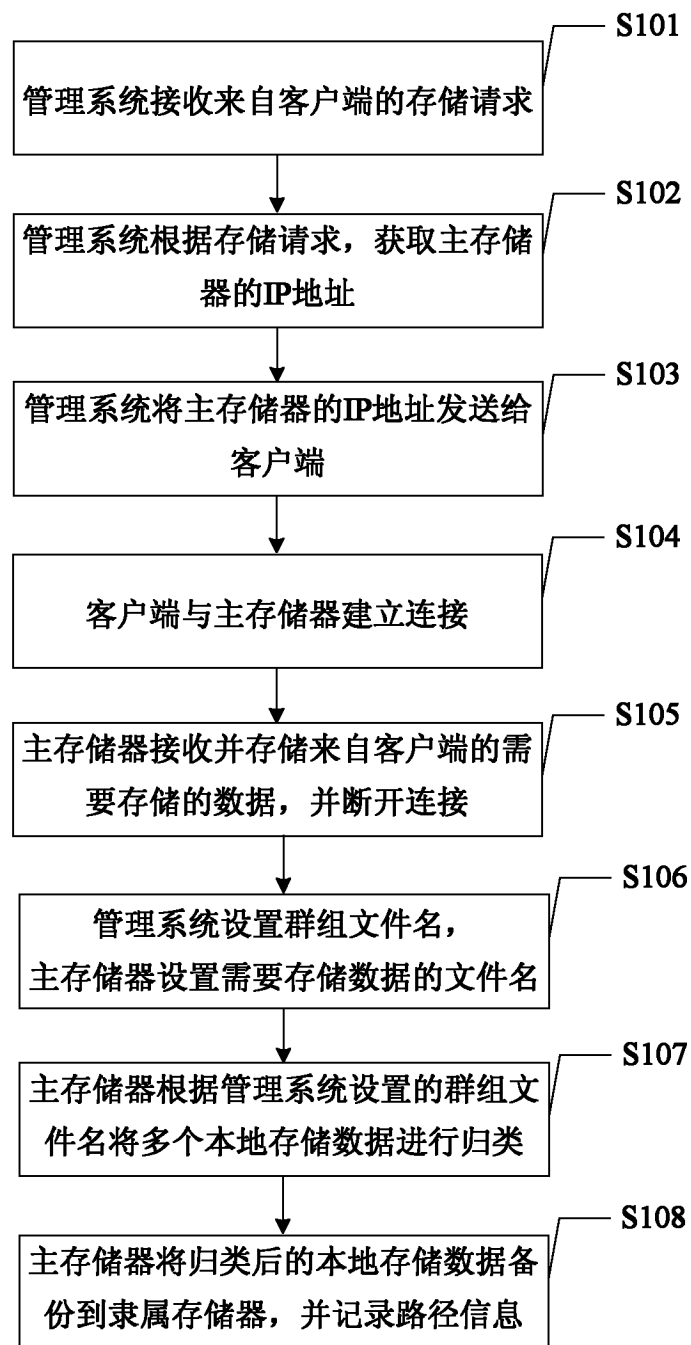


图 1

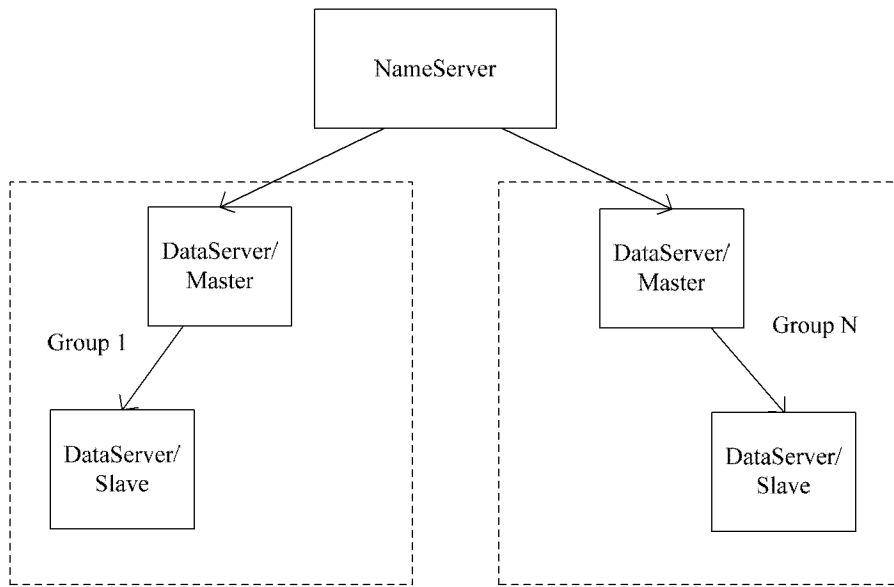


图 2

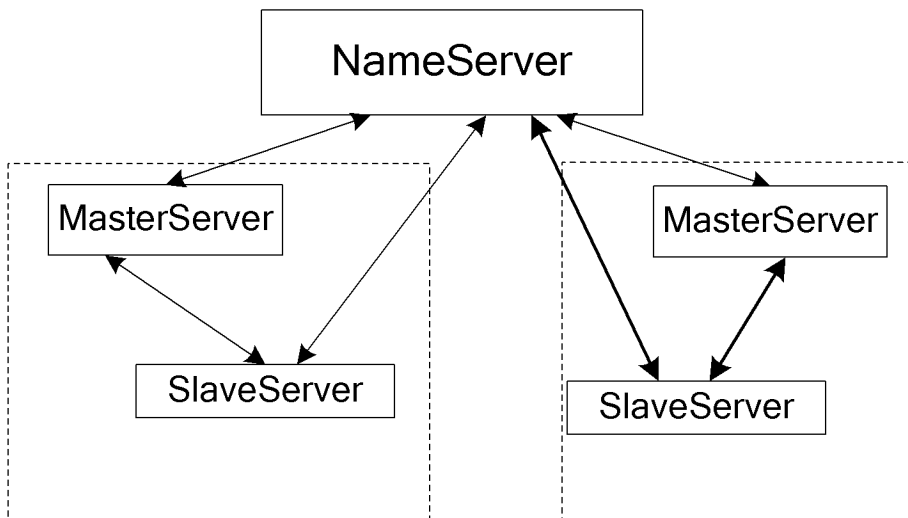


图 3

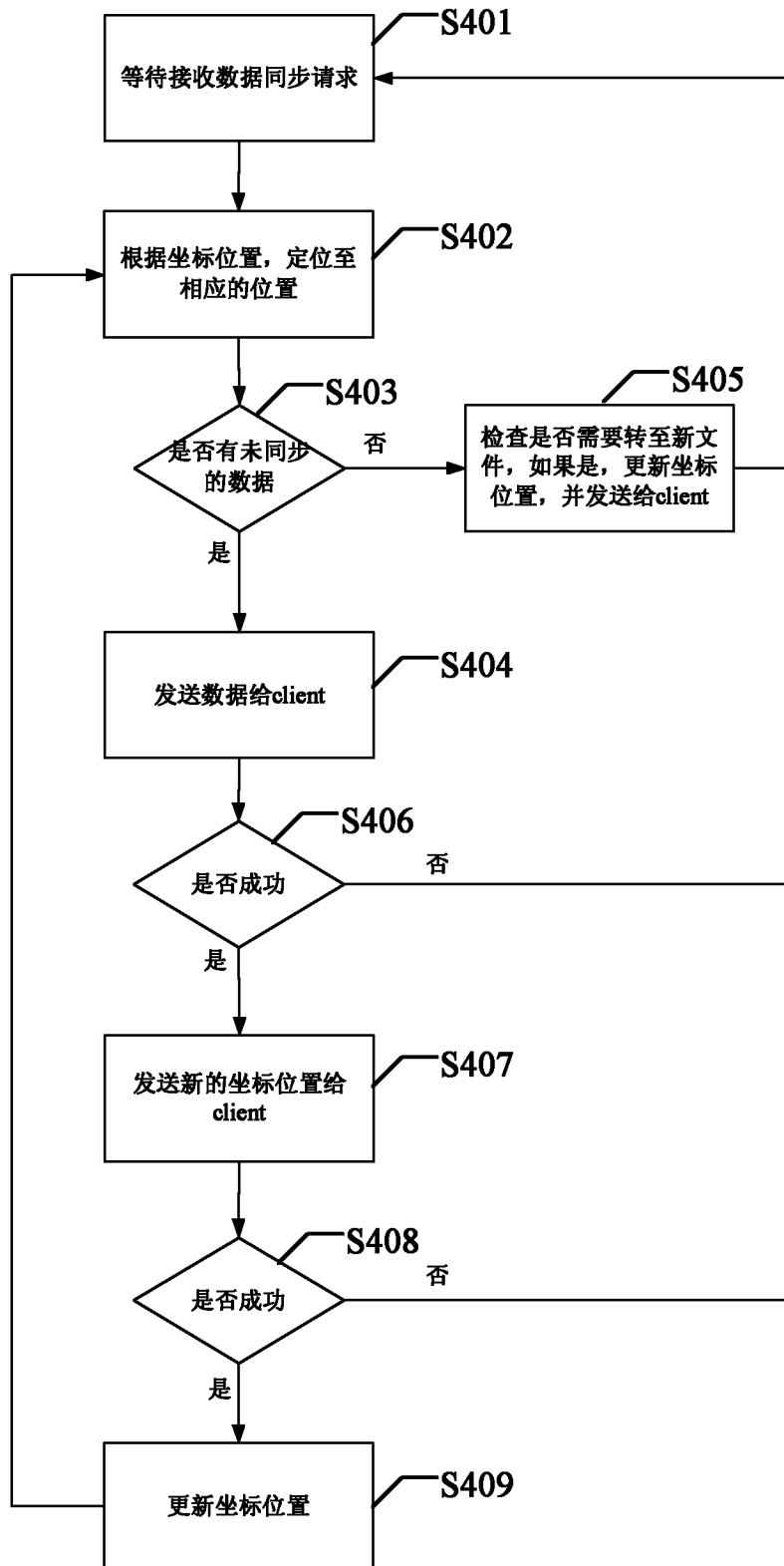


图 4

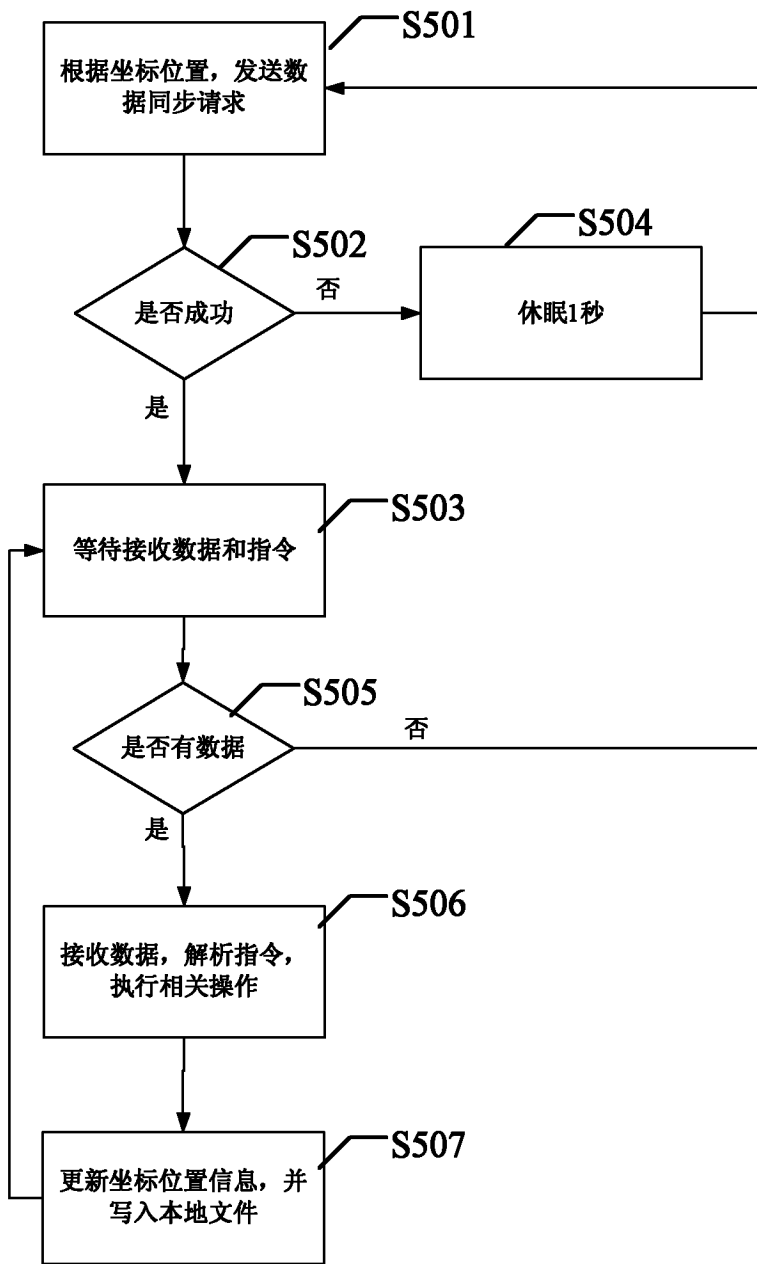


图 5

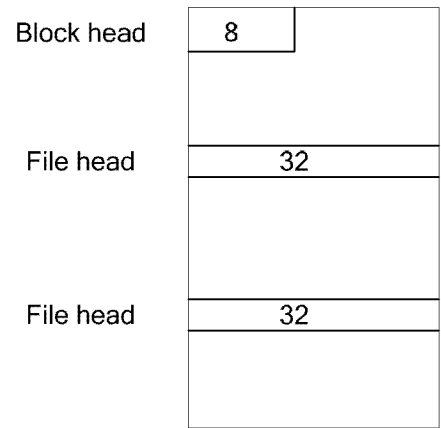


图 6



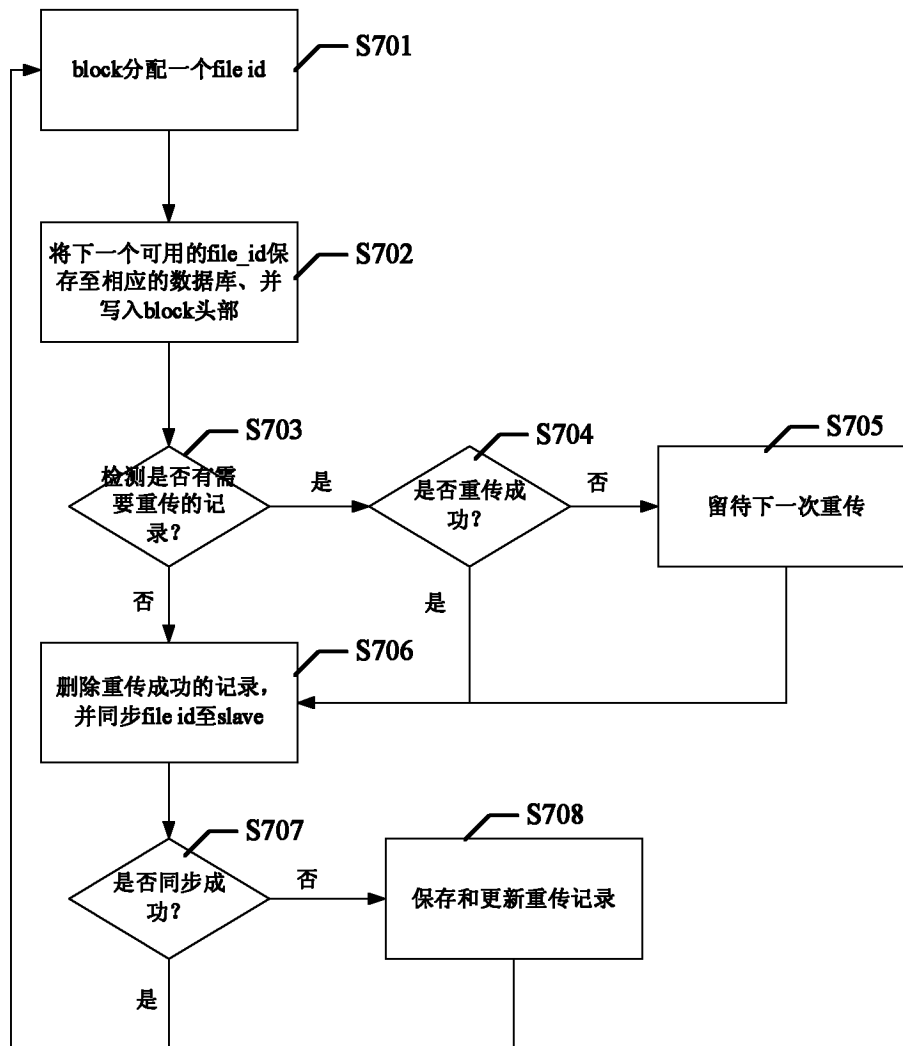


图 7

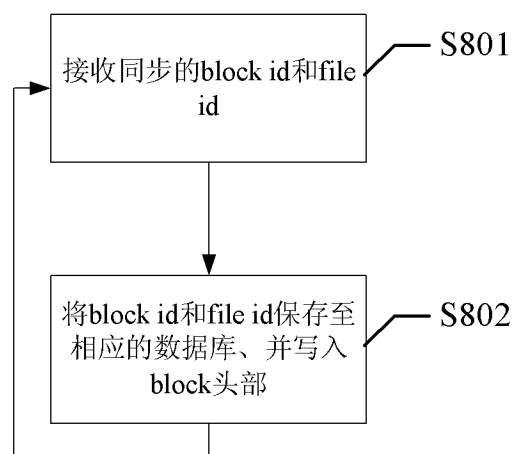


图 8

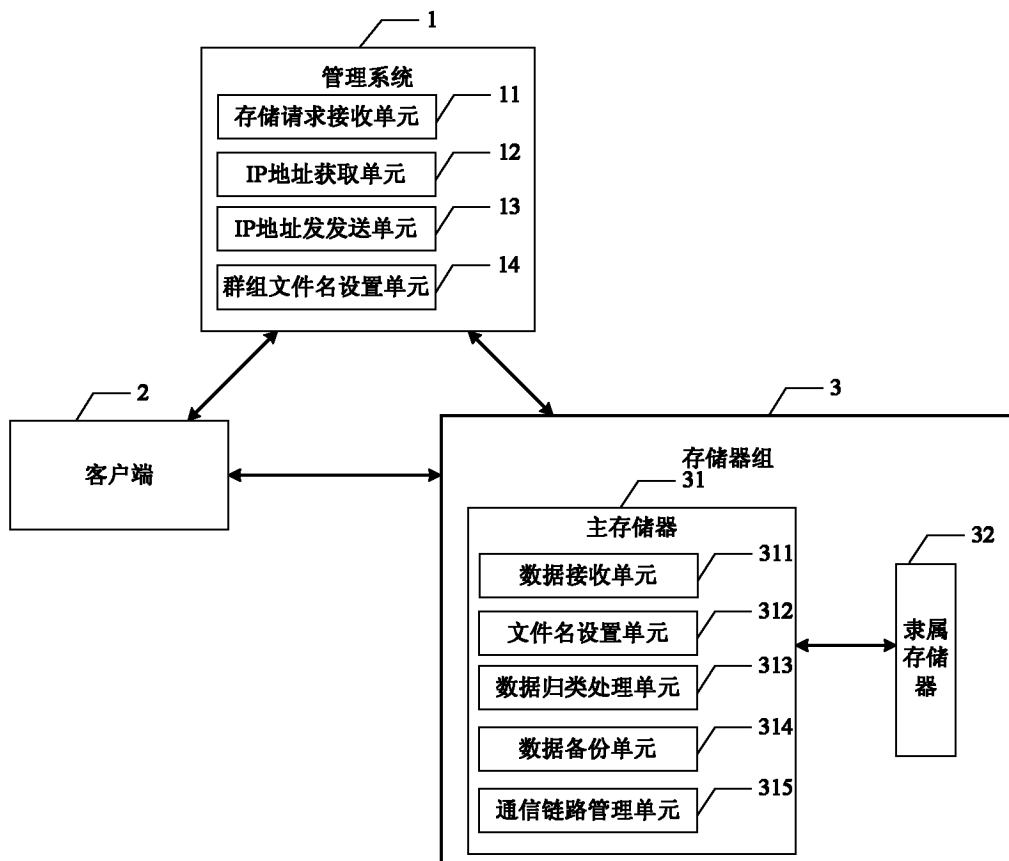


图 9

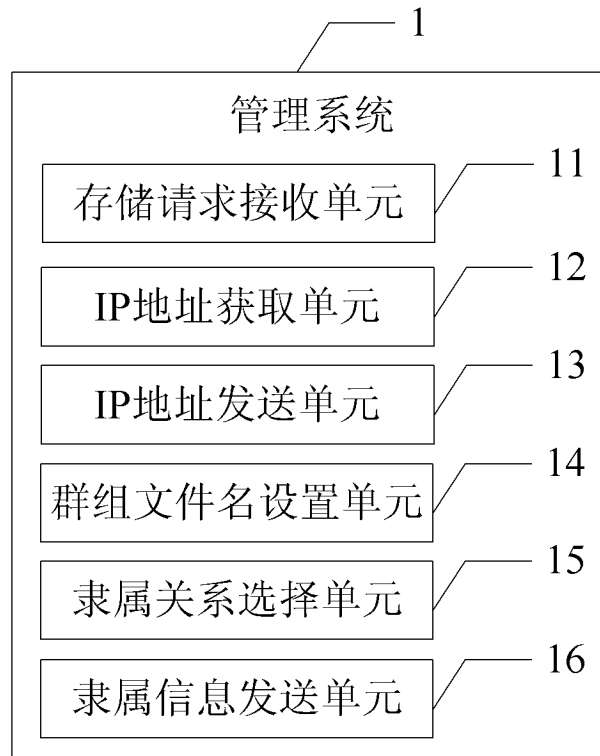


图 10

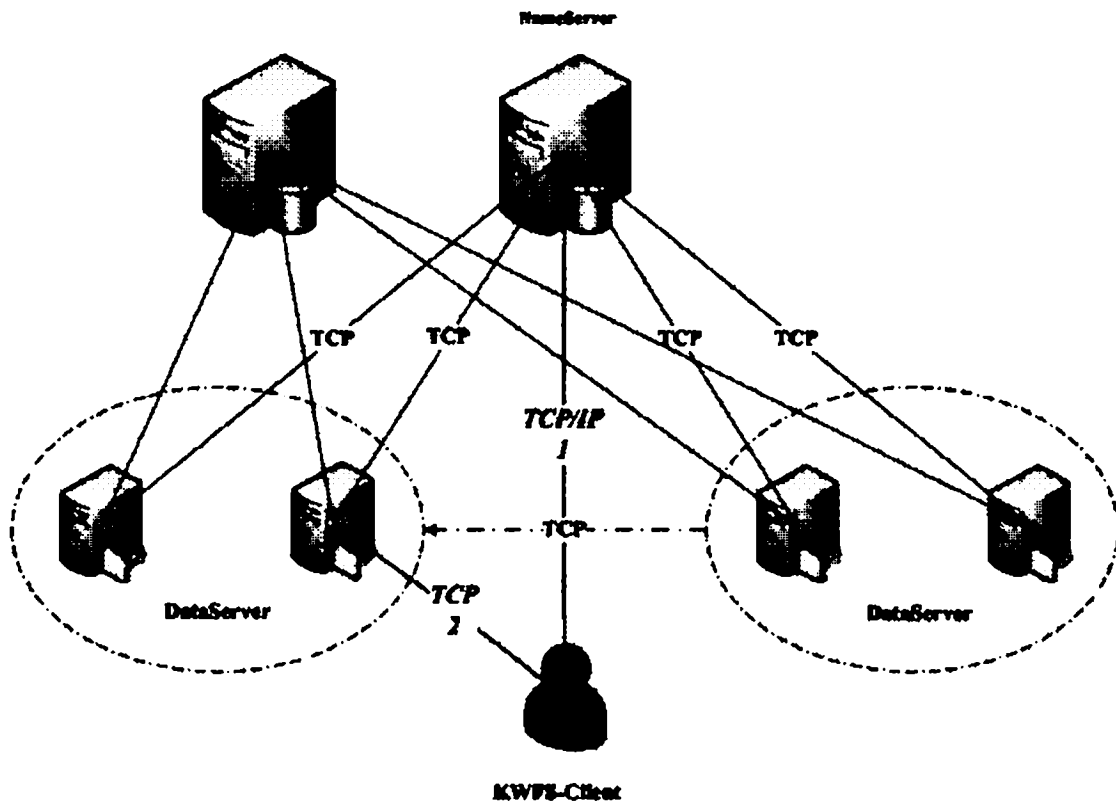


图 11

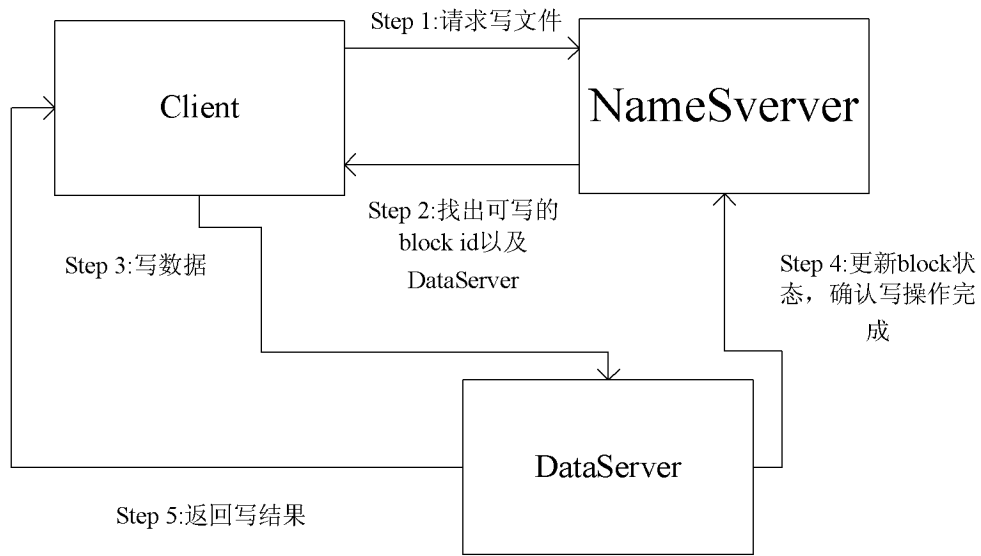


图 12

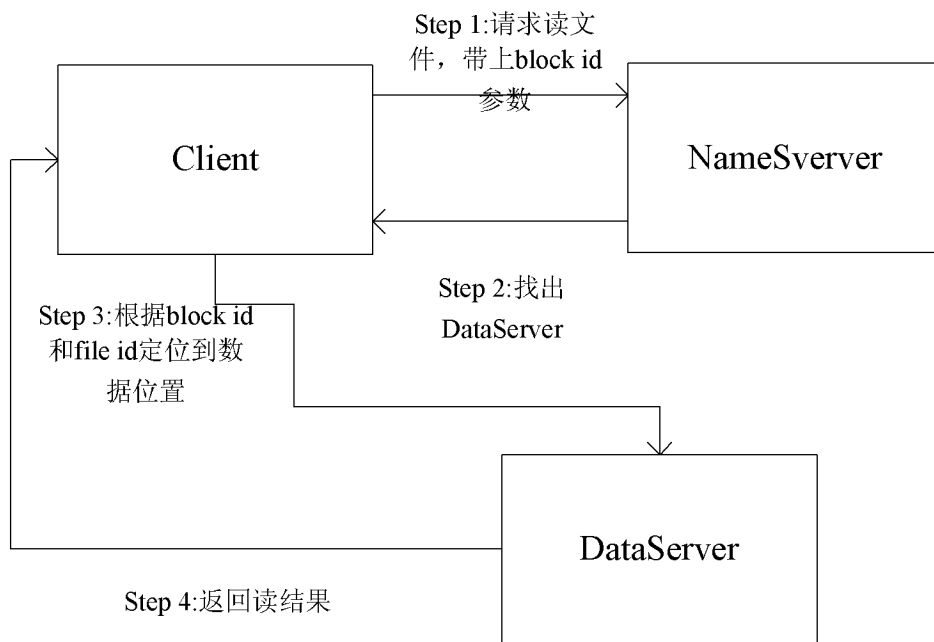


图 13