(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 112860480 A
(43)申请公布日 2021.05.28

(21)申请号 202110080368.3

(22)申请日 2020.12.30

(66)本国优先权数据
202010955301.5 2020.09.11 CN

(62)分案原申请数据
202011628940.7 2020.12.30

(71)申请人 华为技术有限公司
地址 518129 广东省深圳市龙岗区坂田华为总部办公楼

(72)发明人 杜翔 翁宇佳 李小华 张鹏

(51)Int.Cl.
*G06F 11/14*(2006.01)
*G06F 11/20*(2006.01)

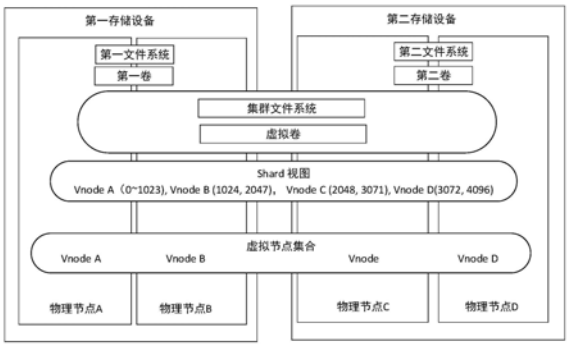
权利要求书3页 说明书14页 附图12页

(54)发明名称

一种双活存储系统及其处理数据的方法

(57)摘要

本申请实施例提供一种双活存储系统及基于所述双活存储系统对数据处理的方法。所述双活存储系统包括第一存储设备和第二存储设备。所述第一存储设备用于接收客户端集群发送给文件系统的第一文件的数据，存储所述第一文件的数据，并且将所述第一文件的数据的第一副本数据发送给所述第二存储设备。所述第二存储设备用于接收所述客户端集群发送给所述文件系统的第二文件的数据，存储所述第二文件的数据，并且将所述第二文件的第二副本数据发送给所述第一存储设备。由于本申请实施例中第一存储设备和第二存储设备都可以通过同样的文件系统进行文件数据的存储，并可以备份对端的文件的数据，从而实现了Act ive-Act ive模式双活存储系统。




CN 112860480 A

1.一种双活存储系统,其特征在于,包括多个虚拟节点,每个虚拟节点的计算资源来源于所述双活存储系统的第一存储设备或第二存储设备中的物理节点,所述双活存储系统的文件系统的数据分布在所述多个虚拟节点对应的物理节点中,并由所述多个虚拟节点分别处理,且第一存储设备与第二存储设备互为备份存储设备。

2.根据权利要求1所述的系统,其特征在于,还包括管理设备,所述管理设备还用于创建全局视图,所述全局视图用于记录每个虚拟节点与其分配的计算资源之间的对应关系;

所述管理设备还用于将所述全局视图发送给所述第一存储设备及所述第二存储设备;

所述第一存储设备还用于保存所述全局视图;

所述第二存储设备还用于保存所述全局视图。

3.根据权利要求2所述的系统,其特征在于,所述第一存储设备用于:

根据接收的所述文件系统的第一文件的数据的地址确定所述第一文件对应的第一虚拟节点;

根据所述第一虚拟节点以及所述全局视图确定为所述第一虚拟节点分配的计算资源;

基于为所述第一虚拟节点分配的计算资源,将所述第一文件的数据发送给所述计算资源对应的物理节点,由所述物理节点将所述第一文件的数据存储至所述物理节点的内存中。

4.根据权利要求3所述的系统,其特征在于,所述第一虚拟节点具有至少一个备份虚拟节点,所述第一虚拟节点对应的物理节点与所述备份虚拟节点对应的物理节点位于不同的存储设备中;

所述第一存储设备还用于:

确定所述第一虚拟节点对应的备份虚拟节点;

根据所述备份虚拟节点以及所述全局视图确定所述备份虚拟节点对应的物理节点;

将所述第一文件的数据的副本数据发送给所述备份虚拟节点对应的物理节点,由所述备份虚拟节点对应的物理节点将所述备份数据存储在所述物理节点中。

5.根据权利要求4所述的系统,其特征在于,所述每个虚拟节点设置有一个或多个分片标识,所述文件系统数据中的每个目录及文件分配一个分片标识,所述第一存储设备和第二存储设备中的物理节点根据每个目录及文件的分片标识将所述目录和文件分布至所述分片标识所属的虚拟节点对应物理节点中。

6.根据权利要求5所述的系统,其特征在于,所述第一存储设备中的第一物理节点用于接收所述第一文件的创建请求,从为所述第一物理节点对应的虚拟节点设置的一个或多个分片标识中为所述第一文件选择一个分片标识,在所述存储设备中创建所述第一文件。

7.根据权利要求1-6任意一项所述的系统,其特征在于,当所述第二存储设备故障或所述第一存储设备与所述第二存储设备之间的链路断开后,所述第一存储设备还用于基于所述第二存储设备中存储的所述第一存储设备的备份数据恢复分布在所述第一存储设备中的物理节点中的所述文件系统的数据,并接管客户端集群发送给所述第二存储设备的业务。

8.根据权利要求7所述的系统,其特征在于,所述第一存储设备还用于从所述全局视图中删除所述第二存储设备的计算资源对应的虚拟节点。

9.根据权利要求1-8任意一项所述的系统,其特征在于,所述第一存储设备还具有第一

文件系统，所述第二存储设备还具有第二文件系统。

10.一种数据处理方法，应用于双活存储系统，所述双活存储系统每个虚拟节点的计算资源来源于所述双活存储系统的第一存储设备或第二存储设备中的物理节点，且所述第一存储设备与所述第二存储设备互为备份存储设备，其特征在于，所述方法包括：

所述第一存储设备接收发送给所述双活存储系统的文件系统的第一文件的数据；

确定所述第一文件的数据对应的虚拟节点，并确定所述虚拟节点的计算资源所在的物理节点，存储所述第一文件的数据至所述物理节点；

存储所述第一文件的数据的副本数据至所述物理节点所在的存储设备的备份存储设备中。

11.根据权利要求10所述的方法，其特征在于，所述双活存储系统还包括管理设备，所述方法还包括：

所述管理设备创建全局视图，所述全局视图用于记录每个虚拟节点与其分配的计算资源之间的对应关系；

所述管理设备将所述全局视图发送给所述第一存储设备及所述第二存储设备；

所述第一存储设备及所述第二存储设备保存所述全局视图。

12.根据权利要求11所述的方法，其特征在于，

所述第一存储设备根据所述第一文件的数据的地址确定所述第一文件对应的虚拟节点；

所述第一存储设备在确定所述虚拟节点的计算资源所在的物理节点时，包括：

根据所述虚拟节点以及所述全局视图确定为所述虚拟节点分配的计算资源；

基于为所述虚拟节点分配的计算资源，将所述第一文件的数据发送给所述计算资源对应的物理节点，由所述物理节点将所述第一文件的数据存储至所述物理节点的内存中。

13.根据权利要求12所述的方法，其特征在于，所述第一虚拟节点具有至少一个备份虚拟节点，所述第一虚拟节点对应的物理节点与所述备份虚拟节点对应的物理节点位于不同的存储设备中；

所述方法还包括：

所述第一存储设备确定所述虚拟节点对应的备份虚拟节点；

所述第一存储设备根据所述备份虚拟节点以及所述全局视图确定所述备份虚拟节点对应的物理节点；

所述第一存储设备将所述第一文件数据的副本数据发送给所述备份虚拟节点对应的物理节点，由所述备份虚拟节点对应的物理节点将所述第一备份数据存储在所述物理节点中。

14.根据权利要求10-13任意一项所述的系统，其特征在于，所述文件系统所包括的文件和目录分布在所述虚拟节点集合中的多个虚拟节点对应的物理节点中。

15.根据权利要求14所述的方法，其特征在于，所述每个虚拟节点设置有一个或多个分片标识，所述文件系统中的每个目录及文件分配一个分片标识，所述第一存储设备和第二存储设备中的物理节点根据每个目录及文件的分片标识将所述目录和文件分布至所述分片标识所属的虚拟节点对应物理节点中。

16.根据权利要求15所述的方法，其特征在于，所述方法还包括：

所述第一存储设备中的第一物理节点接收所述第一文件的创建请求，从为所述第一物理节点对应的虚拟节点设置的一个或多个分片标识中为所述第一文件选择一个分片标识，在所述第一存储设备中创建所述第一文件。

17.根据权利要求10-16任意一项所述的方法，其特征在于，所述方法还包括：当所述第二存储设备故障或与所述第一存储设备与所述第二存储设备之间的链路断开后，所述第一存储设备基于所述第二存储设备中存储的所述第一存储设备的备份数据恢复分布在所述第一存储设备中的物理节点中的所述文件系统的数据，并接管客户端集群发送给所述第二存储设备的业务。

18.根据权利要求17所述的方法，其特征在于，所述方法还包括所述第一存储设备从所述全局视图中删除所述第二存储设备的计算资源对应的虚拟节点。

19.根据权利要求10-18任意一项所述的方法，其特征在于，所述第一存储设备还具有第一文件系统，所述第二存储设备还具有第二文件系统。

# 一种双活存储系统及其处理数据的方法

## 技术领域

[0001] 本申请涉及存储领域,特别涉及一种双活存储系统及其处理数据的方法。

## 背景技术

[0002] 对于网络存储集群,例如网络附加存储(Network Attached Storage,NAS)集群,在实现双活时,其中的第一存储设备在接收到写入数据时,会将所接收到的写入数据写到本地的同时,会同步到对端存储设备作为备份数据,这样,在第一存储设备故障,或者第一存储设备与第二存储设备断开连接时,第二存储设备可利用所述备份数据接管第一存储设备的业务,从而保证业务不中断,即实现了Active-Passive模式的双活。但是无法实现Active-Active模式的双活。

## 发明内容

[0003] 本申请提供一种双活存储系统及实现双活存储系统的方法,用于实现Active-Active模式的双活,使双活存储系统中的存储设备可以访问同一文件系统中的数据。

[0004] 本申请第一方面提供一种双活存储系统。所述双活存储系统包括第一存储设备和第二存储设备。所述第一存储设备用于接收客户端集群发送给文件系统的第一文件的数据,存储所述第一文件的数据,并且将所述第一文件的数据的第一副本数据发送给所述第二存储设备。所述第二存储设备用于接收所述客户端集群发送给所述文件系统的第二文件的数据,存储所述第二文件的数据,并且将所述第二文件的第二副本数据发送给所述第一存储设备。

[0005] 由于第一存储设备和第二存储设备都可以通过同样的文件系统进行文件数据的存储,并可以备份对端的文件数据,从而实现了Active-Active模式双活存储系统。传统的NAS设备也具有文件系统,但Active-Passive模式下的两个存储设备各自拥有独立的文件系统,两个独立的文件系统都需要占用存储设备的计算/存储资源,使得资源的利用效率低,管理起来也较为复杂,这并非真正意义上双活。在本申请中,第一存储设备和第二存储设备拥有同一个文件系统,可以提高资源的利用效率,降低了管理复杂度。另外,由于客户端给存储设备发送访问请求时,对它而言也是向同一个文件系统发送请求。因此对客户端而言其访问效率也提高了。

[0006] 在本申请第一方面的一种可能的实现中,所述双活存储系统还包括虚拟节点集合,所述虚拟节点集合包括多个虚拟节点,每个虚拟节点分配有计算资源,所述计算资源来源于所述第一存储设备或所述第二存储设备中的物理节点。

[0007] 所述物理节点可以为所述第一存储设备及所述第二存储设备控制节点,也可以是控制节点中的CPU,或者CPU中的内核。虚拟节点是逻辑上的概念,它作为资源分配的媒介用于实现该系统中计算资源的隔离。按照这种资源管理方式,各个虚拟节点分配有独立的计算资源,那么对应不同虚拟节点的文件/目录所使用的计算资源也是独立的。由此有利于所述双活存储系统的扩容或减容,也有利于实现计算资源之间的免锁机制,降低了复杂度。

[0008] 在本申请第一方面的一种可能的实现中，所述双活存储系统还包括管理设备，所述管理设备还用于创建全局视图，所述全局视图用于记录每个虚拟节点与其分配的计算资源之间的对应关系；所述管理设备还用于将所述全局视图发送给所述第一存储设备及所述第二存储设备，所述第一存储设备及所述第二存储设备保存所述全局视图。

[0009] 所述管理设备可以作为一个软件模块，安装在第一存储设备或第二存储设备上，也可以为一个独立的设备，在作为安装在第一存储设备上的软件模块时，在生成全局视图后，通过与存储设备中的其他模块的交互，将所述全局视图发送至第一存储设备及所述第二存储设备存储。

[0010] 通过全局视图的方式将虚拟节点集合中的虚拟节点分别呈现给第一存储设备及第二存储设备中的应用，第一存储设备及第二存储设备中的应用会将对端的物理节点作为本端的资源进行使用，从而更方便与对端物理节点进行交互。

[0011] 在本申请第一方面的一种可能的实现中，在存储所述第一文件的数据时，所述第一存储设备根据所述第一文件的数据的地址确定所述第一文件对应的第一虚拟节点，根据所述第一虚拟节点以及所述全局视图确定为所述第一虚拟节点分配的计算资源，并基于为所述第一虚拟节点分配的计算资源，将所述第一文件的数据发送给所述计算资源对应的物理节点，由所述物理节点将所述第一文件的数据存储至所述物理节点的内存中。

[0012] 通过全局视图提供的虚拟节点集合，第一存储设备可以接收归属于所述虚拟节点集合中的任意虚拟节点对应的物理节点的文件的数据，并将所接收的文件的数据转发至所述文件归属的物理节点进行处理，这样，用户在写数据的时候，不必感知文件实际存储的位置，可通过任意一个存储设备对文件进行操作。

[0013] 在本申请第一方面的一种可能的实现中，所述第一虚拟节点具有至少一个备份虚拟节点，所述第一虚拟节点对应的物理节点与所述备份虚拟节点对应的物理节点位于不同的存储设备中，在确定第一文件对应的第一虚拟节点后，所述第一存储设备还会确定所述第一虚拟节点对应的备份虚拟节点；根据所述备份虚拟节点以及所述全局视图确定所述备份虚拟节点对应的物理节点；将所述第一副本数据发送给所述备份虚拟节点对应的物理节点，由所述备份虚拟节点对应的物理节点将所述第一备份数据存储在所述物理节点中。

[0014] 通过将写入第一存储设备的文件的数据备份到第二存储设备中，在第一存储设备故障或者与第二存储设备断开连接后，可通过所述备份数据接管第一存储设备的业务，从而提升了系统的可靠性。

[0015] 在本申请第一方面的一种可能的实现中，所述文件系统所包括的文件和目录分布在所述虚拟节点集合中的多个虚拟节点对应的物理节点中。

[0016] 所述文件系统所包括的文件和目录分布在所述虚拟节点集合中的多个虚拟节点对应的物理节点中，具体指将所述文件系统所包括的文件和目录打散给多个物理节点处理。这样，可以充分利用第一存储设备和第二存储设备的物理资源，提高文件的处理效率。

[0017] 在本申请第一方面的一种可能的实现中，所述虚拟节点集合中的每个虚拟节点设置有一个或多个分片标识，所述文件系统中的每个目录及文件分配一个分片标识，所述第一存储设备和第二存储设备中的物理节点根据每个目录及文件的分片标识将所述目录和文件分布式至所述分片标识所属的虚拟节点对应物理节点中。

[0018] 通过分片标识可以更方便的将所述文件系统所包括的文件和目录分布至第一存

储设备及第二存储设备的各个物理节点。

[0019] 在本申请第一方面的一种可能的实现中，所述第一存储设备中的第一物理节点用于接收所述第一文件的创建请求，从为所述第一物理节点对应的虚拟节点设置的一个或多个分片标识中为所述第一文件选择一个分片标识，在所述存储设备中创建所述第一文件。

[0020] 在创建文件时，通过给文件分配接收到所述文件创建请求对应的物理节点的虚拟节点的分片标识，避免将所述文件创建请求转发至其他物理节点，从而提高了处理效率。

[0021] 在本申请第一方面的一种可能的实现中，当所述第二存储设备故障或所述第一存储设备与所述第二存储设备之间的链路断开后，所述第一存储设备还用于基于所述第二文件的第二副本数据恢复所述第二文件，并接管所述客户端集群发送给所述第二存储设备的业务。

[0022] 在第一存储设备故障或者与第二存储设备断开连接后，可通过备份数据接管第一存储设备的业务，从而提升了系统的可靠性。

[0023] 在本申请第一方面的一种可能的实现中，所述第一存储设备还用于从所述全局视图中删除所述第二存储设备的计算资源对应的虚拟节点。

[0024] 在本申请第一方面的一种可能的实现中，所述第一存储设备还具有第一文件系统，所述第二存储设备还具有第二文件系统。

[0025] 在同一存储设备中同时运行本地文件系统及集群的文件系统，为用户提供了多种访问存储设备中的数据的方式。

[0026] 本申请第二方面提供一种实现双活文件系统的方法，所述方法所包括的步骤用于实现本申请第一方面提供的双活存储系统中第一存储设备及第二存储设备所执行的各个功能。

[0027] 本申请第三方面提供一种管理设备，所述管理设备用于创建全局视图，所述全局视图用于记录每个虚拟节点与其分配的计算资源之间的对应关系，还用于将所述全局视图发送给所述第一存储设备及所述第二存储设备存储。

[0028] 所述管理设备用于监控第一存储设备及第二存储设备中的虚拟节点的变化，当侦测到所述虚拟集群中增加了新的虚拟节点，或者当虚拟节点被删除，例如虚拟节点对应的物理节点故障，则更新所述全局视图。

[0029] 通过所述监控模块可以实时监控虚拟节点集群中的虚拟节点的变化，从而及时更新所述全局视图。

[0030] 本申请第四方面提供一种存储介质，用于存储程序指令，所述程序指令用于实现第三方面提供的管理设备提供的各项功能。

## 附图说明

[0031] 为了更清楚地说明本申请实施例或中的技术方案，下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍。

[0032] 图1为Active-passive模式的双活存储系统的架构图。

[0033] 图2为本申请实施例提供的Active-Active模式的双活存储系统的架构图。

[0034] 图3A为本申请实施例中建立双活存储系统的方法的流程图。

[0035] 图3B为本申请实施例中构建双活存储系统的过程中所生成各项参数的示意图。

[0036] 图4A为本申请实施例建立所述双活存储系统的文件系统的流程图。

[0037] 图4B为本申请实施例所构建的双活系统的示意图。

[0038] 图5为本申请实施例在文件系统中创建目录的方法的流程图。

[0039] 图6为本申请实施例在文件系统中查询目录的方法的流程图。

[0040] 图7为本申请实施例在文件系统中创建文件的方法的流程图。

[0041] 图8为本申请实施例在文件系统中的文件中写入数据的方法的流程图。

[0042] 图9为本申请实施例在文件系统中的文件中写入数据的方法的流程图。

[0043] 图10为本申请实施例中第一存储设备接管所述第二存储设备的业务的示意图。

[0044] 图11为本申请实施例中第一存储设备接管所述第二存储设备的业务的方法的流程图。

## 具体实施方式

[0045] 下面将结合本申请实施例中的附图，对本申请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申请一部分实施例，而不是全部的实施例。

[0046] 如图1所示，为Active-passive模式的双活系统的架构示意图。所述系统10包括第一存储设备100及第二存储设备200。在第一存储设备100的控制节点101中(第一存储设备可包括多个控制节点，为方便描述，仅以一个为例进行说明)设置有第一文件系统102，在第二存储设备200的控制节点201(第二存储设备也可包括多个控制节点，为方便描述，仅以一个为例进行说明)中设置有第二文件系统202。当第一客户端300连接至第一存储设备100之后，所述第一存储设备100将所述第一文件系统102挂载至所述第一客户端300。当第二客户端400连接至第二存储设备200之后，所述第二存储设备200将所述第二文件系统202挂载至所述第二客户端400。每个文件系统都具有根目录，所述存储设备将文件系统挂载至所述客户端是指存储设备将文件系统的根目录提供给客户端，客户端将文件系统的根目录设置在客户端的文件系统中，从而使客户端可以获取所述存储设备的文件系统的根目录，从而根据所述存储设备的文件系统的根目录访问存储设备的文件系统。如此，在第一文件系统102挂载至所述第一客户端300之后，第一客户端300通过第一文件系统102进行数据的读写，写入的数据作为本端数据103存储。另外，在第一存储设备100中还会存储第二存储设备200的备份数据，即对端备份数据104。同理，第二客户端400通过第二文件系统202进行数据的读写，写入的数据作为本端数据203存储。另外，在第二存储设备200中还会存储第一存储设备100的备份数据，即对端备份数据204。这样，在第一存储设备100发生故障或者与第二客户端之间的链接断开后，第二客户端可以利用对端备份数据204接管第一客户端300的业务，即实现Active-passive模式的双活。但是在Active-passive模式的双活系统100中，在第一存储设备100和第二存储设备200都正常运行期间，第一客户端300只能通过第一文件系统访问第一存储设备100中的数据，而无法访问第二存储设备200中的数据，第二客户端400只能通过第二文件系统访问第二存储设备200中的数据，而无法访问第一存储设备100中的数据，即无法实现Active-Active模式的双活。

[0047] 本申请实施例提供的技术方案通过设置全局视图，全局视图为虚拟节点的集合，全局视图中的每个虚拟节点分配有计算资源，所述计算资源来自于所述第一存储设备和所述第二存储设备的物理节点，所述物理节点可以是第一存储设备中的控制器及第二存储设

备的控制器，也可以是控制器中的CPU，或者CPU中的内核，还可以是分布式存储系统中的服务器。在本申请实施例中，每个物理节点都可以获取所述全局视图，另外，每个物理节点还使用相同的文件系统，这样，连接至第一存储设备的第一客户端和连接至第二存储设备的第二客户端挂载有相同的文件系统，如此，所述第一客户端可以通过所述文件系统及所述全局视图访问所述第二存储设备中归属于所述文件系统的数据。下面将结合附图对本申请实施例提供的方案进行详细描述。

[0048] 如图2所示，为本申请实施例提供的Active-Active模式的双活系统500的架构图。所述系统500包括第一存储设备600及第二存储设备700。所述第一存储设备600包括物理节点A及物理节点B。第二存储设备700包括物理节点C及物理节点D。在实际应用中，所述第一存储设备600及第二存储设备700可以包括更多的物理节点，为了方便描述，本实施例仅以每个存储设备包括两个物理节点为例进行说明。所述第一存储设备600和第二存储设备700分别包括由多个存储盘构成的持久性存储设备601及701，用于持久性存储数据。基于所述持久性存储设备601及701的存储盘提供的物理存储空间，第一存储设备600及第二存储设备700分别创建第一卷609及第二卷703。第一存储设备600及第二存储设备700可分别根据所述第一卷609及第二卷703将数据存储至持久性存储设备601及701中。所述存储盘例如可以为固态硬盘(Solid State Disk,SSD)，硬盘驱动器(Hard Disk Drive,HDD)等持久性存储介质

[0049] 物理节点A、物理节点B、物理节点C、及物理节点D的结构相同，本申请实施例中仅以节点A的结构为例进行说明。物理节点A包括处理器602及内存603。内存603中存储应用程序指令(图未示)及处理器运行过程中产生的数据。所述处理器602执行所述应用程序指令以实现本申请实施例提供的Active-Active模式的双活功能。所述内存603中除了第一文件系统608之外，还存储有全局视图604、文件系统605、缓存数据606及备份数据607。第一文件系统608的功能与图1中的第一文件系统102的功能相同，在此不再赘述。即在本申请实施例中，每个物理节点包括两个文件系统，一个是各个物理节点共有的文件系统，另外一个是每个物理节点自己的文件系统。关于内存603中其他数据的详细介绍结合实现双活的方法，例如图XX到图XX所示的流程图进行介绍。第一客户端800连接至所述第一存储设备600，以访问第一存储设备600中的数据，第二客户端700连接至所述第二存储设备900，以访问第二存储设备900中的数据。

[0050] 下面将结合流程图3A，图4A，图5-图9介绍本申请实施例实现Active-Active模式的双活的方法。

[0051] 首先，如图3A所示，为本申请实施例提供的建立全局视图的方法的流程图。

[0052] 步骤S301，第一存储设备600的物理节点A接收客户端发送的虚拟集群建立请求。

[0053] 当需要构建双活系统时，会建立全局视图，用户可通过客户端发送全局视图建立请求至第一存储设备600。第一存储设备为主阵列，第一存储设备600中的物理节点A为主节点，则由所述物理节点A对所述请求进行处理。

[0054] 步骤S302，物理节点A建立全局视图604，并同步所建立的全局视图604至全局视图中的其他虚拟节点对应的物理节点。

[0055] 在第一存储设备600与第二存储设备700建立网络连接后，第一存储设备600会获取第二存储设备700中各物理节点的标识，及每个物理节点的IP地址。在建立所述全局视图

604时，所述节点A为第一存储设备600及第二存储设备700中的每个物理节点分配虚拟标识，以标识虚拟节点，并建立全局视图记录所述虚拟节点的虚拟标识。每个物理节点的计算资源，例如处理器资源及内存资源即为所述虚拟节点分配的计算资源，在其他实施例中，除了所述计算资源，还可以给每个虚拟节点分配其他的物理资源，例如带宽等。在本申请实施例中，各个虚拟节点所分配的物理资源是相互独立的，这样，对于一个存储设备来说，可以更方便的对存储设备进行扩容，例如，当个存储设备中增加了新的物理资源，根据新的物理资源生成新的虚拟节点，从而增加虚拟节点的数量，并将新添加的虚拟节点加入所述全局视图。在分布式的存储里，所增加的服务器作为新增的物理资源，根据所增加的服务器建立虚拟节点，从而增加全局视图中虚拟节点的数量。所建立的全局视图如图3B中的Vcluster所示，例如，为第一存储设备600中的物理节点A及物理节点B分配虚拟标识VNode A及Vnode B，为第二存储设备700的物理节点C及物理节点D分配虚拟标识VNode C及VNode D。在生成所述全局视图604后，所述节点A将所述全局视图604存储至内存603及持久性存储设备601中，然后将所述节点集合表604同步至其他虚拟节点对应的物理节点中(物理节点B、C及D)，及第二存储设备700的持久性存储介质701中。

[0056] 步骤S303，物理节点A根据所述节点集合生成分片(shard)视图，并同步所述shard视图至虚拟节点集群中的其他虚拟节点对应的物理节点中。

[0057] 在本申请实施例中，会为所述虚拟集群设置预设数量的Shard，例如4096个，这些shard会均分给所述全局视图604中各个虚拟节点，即生成shard视图。所生产的shard视图如图3B中的shard视图所示。所述shard用于将文件系统605的目录及文件分布存储至所述全局视图604中的各个虚拟节点对应的物理节点中，关于shard视图的具体作用将在下文做详细介绍。在shard视图生成之后，所述物理节点A将所述shard视图存储至本地内存603及持久性存储介质601，并同步所述shard视图至其他虚拟节点对应的物理节点中(物理节点B、C及D)，及第二存储设备700的持久性存储介质701中。

[0058] 步骤S303，物理节点A生成数据备份策略，并同步所述数据备份策略至虚拟节点集群中的其他虚拟节点对应的物理节点中。

[0059] 为了保证数据的可靠性，防止设备故障后数据丢失，本申请实施例可设置数据备份策略，即将所生成的数据备份至多个节点。本申请实施例中的备份策略为对数据进行3副本备份，其中两份存储在本地的两个物理节点中，另外一份存储在远端存储设备的物理节点中。具体地，如图3B所示的备份策略，为每个虚拟节点设置一组备份节点，例如设置虚拟节点Vnode A的对应的备份节点为虚拟节点VnodeB及VnodeC，虚拟节点VnodeB对应的虚拟节点为VnodeA及VnodeD，虚拟节点VnodeB对应的虚拟节点为VnodeA及VnodeD，虚拟节点VnodeD对应的虚拟节点为VnodeC及VnodeB。在备份策略生成之后，所述节点A将所述备份策略存储至本地内存603及持久性存储设备601中，并同步所述备份策略至第二存储设备700的持久性存储设备701及全局视图中的其他虚拟节点对应的物理节点中。

[0060] 图3A虚拟集群的建立由一个管理模块执行，在图3A及图4A中以所述管理模块位于第一存储设备为例进行说明，该管理模块生成所述文件系统及全局视图后，可以将所生成的文件系统及全局视图发送给第一存储设备及第二存储设备进行存储。在其他实施例中，所述管理模块也可以位于一个独立的第三方管理设备上，第三方的管理设备在生成所述文件系统及全局视图后，发送给第一存储设备及第二存储设备中的存储，使各个物理节点都

可以获取所述全局视图。

[0061] 在所建立的虚拟集群运行期间,会通过一个监控模块监控第一存储设备及第二存储设备中的虚拟节点的变化,当侦测到所述虚拟集群中增加了新的虚拟节点,或者当虚拟节点被删除,例如虚拟节点对应的物理节点故障,则所述监控模块会通知所述管理模块更新所述全局视图。所述监控模块可以位于所述第三方管理设备上,也可以位于第一存储设备及第二存储设备中。第一存储设备作为主存储设备,第二存储设备会将监控到的变化发送至第一存储设备,由第一存储设备中的管理模块更新所述全局视图。如此即可完成虚拟节点集群的建立,在虚拟节点集群建立好之后,所述第一存储设备600及所述第二存储设备700即可根据客户端的请求建立文件系统。具体如图4A的流程图所示。

[0062] 步骤S401,物理节点A接收文件系统创建请求。

[0063] 第一客户端800可以向第一存储设备600发送所述文件系统创建请求,也可以向第二存储设备700发送文件系统创建请求,如果是第一存储设备600接收所述文件系统创建请求,则由所述物理节点A处理所述文件系统创建请求,如果是第二存储设备700接收所述文件系统创建请求,则第二存储设备700转发所述文件系统创建请求至所述第一存储设备600的物理节点A处理。

[0064] 步骤S402,物理节点A为所述文件系统设置根目录。

[0065] 所述主节点在设置所述根目录时,首先生成根目录的标记,一般情况下,根目录的默认标记为“/”,接着为所述根目录分配标识信息及shard ID。由于在主节点创建的shard视图同步到了所有节点,所以,主节点从自己的内存中获取所述shard视图,从中为所述根目录选择shard ID。如表3B所示,所述shard视图中每个虚拟节点都被分配了多个shard ID,所以,为了减少跨网络及跨节点的访问,会优先从物理节点A对应的虚拟节点Vnode A所包括的shard ID中为所述根目录分配shard ID。由于是根目录,shard ID还没有被分配过,则例如可以选择shard 0作为所述根目录的shard ID。

[0066] 步骤S403,物理节点A发送文件系统的挂载命令至所述第一客户端800。

[0067] 在所述集群文件系统的根目录生成之后,为了能使第一客户端800访问所述文件系统,物理节点A会将所述文件系统挂载至所述第一客户端800的文件系统中。例如,物理节点A通过mount命令将所述文件系统的根目录提供至第一客户端800,物理节点A在发送mount命令时,会携带所述根目录的参数信息,所述根目录的参数信息即为所述根目录的句柄信息,所述句柄信息中携带了所述根目录的shard ID及标识信息。

[0068] 步骤S404,所述第一客户端800根据所述挂载命令将所述集群文件系统挂载至所述第一客户端800的文件系统中。

[0069] 在第一客户端800收到所述文件系统的根目录参数信息后,会在第一客户端的文件系统上生成一个挂载点,并在所述挂载点处记录所述文件系统的根目录的参数信息,所述挂载点为一段存储空间。

[0070] 如此,第一客户端800除了可以通过第一文件系统608与第一存储设备600进行数据传输外,还可以通过所述文件系统605与第一客户端800进行数据传输。用户可以根据实际需求选择需要访问的文件系统。

[0071] 步骤S405,物理节点A为所述文件系统分配虚拟卷。

[0072] 每个新建的文件系统都会被分配一个虚拟卷Vvloume 0,用于写入第一客户端或

者第二客户端写入该文件系统的数据。步骤S406,物理节点A为所述虚拟卷创建镜像卷对。

[0073] 在虚拟卷Vvloume 0建立之后,所述物理节点A首先会基于所述持久性存储介质601创建一个本地卷,例如图2中的第一卷,然后请求第二存储设备700在第二存储设备700中创建所述第一卷的镜像卷,例如图2中的第二卷文件系统文件系统。

[0074] 步骤S407,物理节点A通过记录所述虚拟卷对应镜像卷对生成刷盘策略。

[0075] 所生成的刷盘策略如图3B所示的刷盘策略所示,文件系统的虚拟卷对应的镜像卷对(第一卷及第二卷)。根据图3B所示的刷盘策略,可以将内存中缓存的所述文件系统的数据分别存储至所述第一存储设备600的持久性存储介质601及第二存储设备700的持久性存储介质701中,从而保证数据的可靠性。具体如何根据所述刷盘策略将内存中的数据写入持久性存储介质601及持久性存储介质701将在图9中做详细描述。

[0076] 在生成刷盘策略后,物理节点A将所述文件系统刷盘策略存储至本地内存603及持久性存储设备601,并将其同步至第二存储设备700的持久性存储设备701及全局视图中的其他虚拟节点对应的物理节点中。

[0077] 通过执行图3A及图4B的方法,即可完成Active-Active的双活文件系统的创建,文件系统创建完成的双活存储系统的示意图如图4B所示,即在第一存储设备和第二存储设备之上生成跨设备的文件系统、虚拟卷、Shard视图,及全局视图。

[0078] 在在Active-Active的双活存储系统创建完成之后,即可基于所述文件系统进行目录及文件的创建与访问。

[0079] 首先,结合图5所示的流程图说明在所述文件系统下创建目录的过程。下面以所述根目录为父目录,将待创建的目录作为所述父目录的子目录进行介绍。本申请实施例中,用户可以通过第一客户端访问第一存储设备创建所述子目录,也可以通过第二客户端访问第二存储设备创建所述子目录。在第一存储设备将所述文件系统挂载至所述第一客户端时,即建立了第一客户端访问所述文件系统的路径,例如第一存储设备通过物理节点A将所述文件系统挂载至所述第一客户端,则第一客户端则会通过所述物理节点A访问所述文件系统。为了实现Active-Active的双活访问,第二存储设备也会将所述述文件系统挂载至所述第二客户端的文件系统,如此则建立了第二客户端访问所述文件系统的路径,第二客户端访问所述文件系统的请求会被发送至挂载所述文件系统的物理节点,例如物理节点C。下面以用于通过第二客户端发送子目录创建请求至第二存储设备为例说明子目录的创建过程。

[0080] 具体创建过程如图5的流程图所示。

[0081] 步骤S501,所述第二客户端发送子目录创建请求至物理节点C。

[0082] 所述物理节点C为第二存储设备700的主节点,也就是挂载所述文件系统至所述第二客户端的节点。所述子目录创建请求包括父目录的参数信息及子目录的名称。

[0083] 步骤S502,物理节点C接收第二客户端发送的创建请求,根据所述创建请求为所述子目录生成参数信息。

[0084] 所述参数信息包括父目录的标识信息及shard ID,所述标识信息用于唯一标识所述子目录,所述标识信息例如为NFS文件系统中的对象ID。在生成Shard ID时,所述物理节点C查找所述shard视图,从所述shard视图内记录的Shard ID中为所述子目录分配一个shard ID,然后在该shard ID所归属的虚拟节点对应的物理节点中创建所述子目录。需要说明的是,每个目录可以分配一个shard ID,但是一个shard ID可以被分配给多个目录。在

本申请实施例中,为了减少数据的转发,会在接收到所述子目录请求的物理节点对应的虚拟节点的shard ID中为所述子目录分配shard ID,即在物理节点C对应的虚拟节点Vnode C中所对应的Shard[2048,3071]中为所述子目录分配shard ID。但在所述虚拟节点Vnode C中shard ID对应的目录超过预设阈值的时候,则会为所述子目录分配其他虚拟节点对应的shard ID。

[0085] 步骤S503,物理节点C创建所述子目录。

[0086] 创建所述子目录包括为所述子目录生成目录入口表(directory entry table,DET)及Inode表。所述目录入口表用于记录所述子目录创建成功之后,所述子目录作为父目录,在其下所建立的子目录或者文件的参数信息,所述参数信息例如包括子目录的名称,目录或者文件的标识信息及shard ID等。

[0087] Inode表用于记录后续在所述子目录中所创建的文件的详细信息,例如文件的文件长度、用户对文件的操作权限、文件的修改时间等信息。

[0088] 步骤S504,物理节点C根据所述父目录的参数信息确定所述父目录所归属的第一存储设备中的物理节点B。

[0089] 所述父目录的参数信息中包括Shard ID,通过在所述Shard视图可以确定所述Shard ID对应的虚拟节点为虚拟节点Vnode B,则进一步根据所述虚拟节点Vnode B确定所述虚拟节点Vnode B对应的物理节点为所述第一存储设备中的物理节点B。

[0090] 步骤S505,物理节点C将所述子目录的参数信息及所述父目录的参数信息发送至所述物理节点B。

[0091] 步骤S506,物理节点B根据所述父目录的参数信息找到所述父目录的目录入口表。

[0092] 具体地,可根据所述父目录的参数信息中的shard ID及父目录名称找到所述父目录。

[0093] 步骤S507,物理节点B将所述子目录的参数信息记录在所述父目录的目录入口表中。

[0094] 步骤S508,物理节点B将所述子目录的参数首先返回给所述物理节点C,物理节点C再将所述子目录的参数返回给所述第二客户端。

[0095] 在文件系统中访问文件,例如文件读取或者文件创建的过程中,由于文件创建在目录下,所述首先要查找到目录,才能进一步对该目录下的文件进行访问,如果所访问的文件在多层目录下,则需要逐层查询目录,直到查找到最底层的目录。例如,对于多层目录filesystem1/user1/favorite,由于根目录的参数信息已记录在所述第一客户端的文件系统中,所以所述客户端首先会根据根目录filesystem1的参数信息查询所述子目录user1的参数信息,即生成查询所述user1的请求,等查询到所述user1的参数信息后,再根据所述user1的参数信息查询所述favorite的参数信息,即生成查询所述favorite的请求。每个层级的目录的参数信息的查询方法相同,下面以上层目录为父目录,待查询的目录为子目录为例,说明一次目录查询的过程。本申请实施例中仍然以第二存储设备的物理节点C接收到查询请求为例进行说明

[0096] 步骤S601,第二客户端发送子目录的查询请求至物理节点C。

[0097] 所述查询请求中携带所述父目录的参数信息及子目录的名称。所述父目录的参数信息例如为所述父目录句柄。在所述父目录为根目录时,则从所述客户端的文件系统中获

取所述根目录的句柄。当所述父目录为不是根目录时，则可通过查询所述父目录的查询请求查询到所述父目录的句柄。

[0098] 所述父目录的句柄中包括所述父目录的标识信息及shardID。

[0099] 步骤S602，物理节点C接收第二客户端发送的查询请求，根据所述查询请求确定所述父目录所归属的物理节点B。

[0100] 物理节点C从所述根目录的参数信息中获取所述根目录的shard ID，根据所述shard ID获取所述父目录所归属的虚拟节点。

[0101] 由于物理节点A将创建的shard视图同步到了所有节点，所以，物理节点C从自己的内存中获取所述shard视图，根据所述父目录的Shard ID确定所述父目录所归属的虚拟节点，再确定所述虚拟节点对应的物理节点。

[0102] 步骤S603，物理节点C将所述父目录的参数及所述子目录的名称发送至所述父目录所在的物理节点B。

[0103] 步骤S604，物理节点B根据所述父目录的参数确定所述父目录的目录入口表。

[0104] 请参考图5的描述，在物理节点B创建所述父目录时，会为所述父目录创建目录入口表，所述目录入口表中记录了在所述父目录下创建的所有子目录的参数信息。

[0105] 步骤S605，物理节点B从所述父目录的目录入口表中获取所述子目录的参数信息。

[0106] 步骤S606，物理节点B将所述子目录的参数信息返回给所述物理节点C。

[0107] 步骤S607，物理节点C将所述子目录的参数信息返回给所述第二客户端。

[0108] 图5及图6以第二客户端访问第二存储设备并在文件系统中创建子目录及查询子目录的为例进行说明，但在实际应用中，第一客户端也可以通过访问所述第一存储设备进行所述子目录的创建及查询。

[0109] 在查询到子目录或者创建了新的子目录后，第一客户端或者第二客户端可以获取所述子目录的参数信息，然后可根据所述子目录的参数信息在所述子目录中创建文件。下面以用户通过第一客户端访问第一存储设备，并在子目录中创建文件的过程进行说明，具体如图7所示。

[0110] 步骤S701，客户端发送文件生成请求至物理节点A。

[0111] 所述文件生成请求携带所述子目录的参数信息及文件名。

[0112] 如图5或图6所述，物理节点A已经将所述子目录的参数信息发送至所述客户端，所以所述客户端需要在所述子目录中创建文件时，可在所述文件查询请求中携带所述子目录的参数信息及所述文件的文件名。

[0113] 步骤S702，物理节点A在接收到所述文件生成请求后，根据所述子目录的参数信息确定所述子目录所归属的物理节点D。

[0114] 确定所述子目录所归属的物理节点D的方式与图6中的步骤S602相同在此不再赘述。

[0115] 步骤S703，物理节点A将所述子目录的参数信息及所述文件名发送至所述物理节点D。

[0116] 步骤S704，所述物理节点D确定所述文件是否已经创建。

[0117] 所述物理节点D根据所述子目录的参数中的Shard ID及子目录名称找到所述子目录，然后找到所述子目录对应的DET，并在所述DET中查找所述文件名，如果存在，则说明有

相同文件名的文件已经创建,则执行步骤S705,如果不存在,则说明可以在所述子目录中创建所述文件,则执行步骤S706。

[0118] 步骤S705,所述节点D发送所述文件名已被创建的反馈至节点A,节点A进一步反馈给第一客户端。

[0119] 第一客户端收到所述反馈消息后,可进一步通过通知消息通知用户,相同文件名的文件已存在,用户可根据该提示信息做进一步操作,例如修改文件名。

[0120] 步骤S706,所述节点D创建所述文件。

[0121] 所述节点D创建文件时,为所述文件设置参数,例如分配shard ID,分配文件标识信息,并将shard ID及文件标识信息添加至所述子目录的DET中。如图5中的步骤S503所述,在创建所述子目录时,会为子目录生成Inode表,所述Inode表用于记录在所述子目录下所生成的文件的信息,所以,在本步骤中,在节点D创建了所述文件后,会将所述文件的信息添加至所述子目录中的Inode中。所述文件信息包括文件长度、用户对文件的操作权限、文件的修改时间等信息。

[0122] 步骤S707,物理节点D反馈所述文件参数。

[0123] 物理节点D首先将所述反馈信息发送至节点A,节点A则进一步将所述反馈信息反馈给第一客户端。

[0124] 在步骤S702中,当所述物理节点A确定所述子目录的归属节点为所述物理节点A时,则由物理节点A执行上述步骤S704至S707。

[0125] 在此需要说明的是,图6中生成的子目录及图7中生成的文件都会根据图3B设定的备份策略备份到对应的备份节点中。

[0126] 在文件创建好后,用户即可在所述文件写入数据。用户可通过连接至第一存储设备的第一客户端及连接至第二存储设备的第二客户端在所述文件中写入数据。下面以用户通过第一客户端访问第一存储设备,对所述文件进行数据写入的过程为例进行说明,具体如图8所示。

[0127] 步骤S801,物理节点A接收对所述文件的写入请求。

[0128] 在本申请实施例中,由于任一节点都存储有文件系统,所以用户可以通过连接至任一节点的客户端访问所述文件系统中的文件。

[0129] 所述写入请求中携带所述文件的地址信息,所述地址信息包括所述文件的参数信息、偏移地址及待写数据。在本申请实施例中,所述文件的参数信息即为所述文件的句柄,包括文件系统标识、文件标识、及shard ID。

[0130] 步骤S802,物理节点A根据所述写入请求确定所述文件的归属节点D。

[0131] 根据所述文件的shard ID确定记录所述文件的归属节点D的方式请参考图6中的步骤S602,在此不再赘述。

[0132] 步骤S803,所述物理节点A将所述写请求转发至所述物理节点D。

[0133] 步骤S804,所述物理节点D将对所述文件系统的访问转换为对所述文件系统对应的虚拟卷的访问。

[0134] 由于每个物理节点中都记录了为所述文件系统创建的虚拟卷,则所述物理节点D将所述写请求中的文件系统的标识替换为所述虚拟卷的标识。

[0135] 步骤S805,所述物理节点D根据所述写请求中的文件标识及shard ID找到所述文

件并更新所述文件的信息。

[0136] 所述物理节点D在根据所述文件标识及shard ID找到所述文件之后，再根据所述文件标识中包含有所述文件的Inode号，在所述Inode中找到所述文件对应的inode项，并在其中记录文件的信息，例如根据写请求中携带的待写数据的长度及偏移地址，更新所述文件的长度及偏移地址并将当前时间记录为文件的更新时间。

[0137] 步骤S806，所述物理节点D根据预设的备份策略，对所述待写数据进行多副本写入。

[0138] 在建立所述虚拟文件集群时，为所述文件系统建立了备份策略，所述备份策略中为每个节点设定了备份节点，例如，根据图3B设置的备份策略，可以确定所述物理节点D的备份节点为物理节点C及物理节点B，则所述物理节点D在将所述待写数据写入本地内存的同时，将所述待写数据发送至物理节点C及物理节点B，由所述物理节点C及所述物理节点B将所述待写数据写入自己的内存中。

[0139] 步骤S807，所述物理节点D在确定多副本写入完成后，返回写请求完成的消息至第一客户端。

[0140] 步骤S808，所述物理节点D对所述待写数据进行持久化存储。

[0141] 如图3B所示的刷盘策略，文件系统的虚拟卷对应了镜像卷对：第一存储设备中的第一卷及第二存储设备中的第二卷。所述物理节点D根据预设的内存淘汰算法，在确定需要将所述待写数据淘汰至持久性存储，也即刷盘时，先根据所述待写数据中的地址中记录的所述虚拟卷从所述刷盘策略中获取所述虚拟卷对应的镜像卷对，即第一存储设备中的第一卷及第二存储设备中的第二卷，然后将所述第二存储设备的内存中的待写数据写入所述持久性存储701中所述第二卷对应的物理空间中，然后将所述待写数据的的内存地址发送至所述第一存储设备中所述物理节点D对应的备份节点B，物理节点B根据所述内存地址将所述物理节点B的内存中存储的待写数据写入第一存储设备的持久性存储601中所述第一卷对应的物理空间中。

[0142] 如图9所示，为本申请实施例中对文件进行读取的方法的流程图。

[0143] 在本申请实施例中，用户同样可以通过任一客户端访问所述文件系统中的文件。本实施例以用户通过第二客户端读取文件为例进行说明。

[0144] 步骤S901，物理节点C接收文件的读请求。

[0145] 所述读请求中携带文件的地址信息，所述地址信息包括文件的参数信息、及偏移地址，所述参数信息即为所述文件的句柄，包括文件系统标识、文件标识、及shard ID。在第二客户端发送所述读请求时，已经根据图6所示的方法获取了文件的参数信息。

[0146] 步骤S902，物理节点C根据所述读请求确定所述文件的归属节点B。

[0147] 关于文件的归属节点B的确认方式请参考图6中的步骤S602的描述，在此不再赘述。

[0148] 步骤S903，物理节点C将所述读请求转发至所述归属节点B。

[0149] 步骤S904，物理节点B将所述读请求对所述文件系统的访问转换为对所述文件系统的虚拟卷的访问。

[0150] 步骤S905，所述物理节点B根据所述读请求中的地址，从所述物理节点B的内存中读取所述文件。

[0151] 步骤S906,所述物理节点B返回所述文件。

[0152] 步骤S907,当所述文件不在所述内存中时,所述物理节点B根据所述刷盘策略中所述虚拟卷对应的第一存储设备中的第一卷,从所述持久性存储601中读取所述文件,并返回给所述物理节点C,所述物理节点C再将所述文件返回给所述第二客户端。

[0153] 本申请实施例中,第一存储设备及第二存储设备在进行文件及目录的访问的时候,都是基于shard ID将访问请求转发至文件及目录的归属节点,这样,会导致跨设备进行数据的访问,从而影响访问效率。在本申请实施例提供的一种可能的实现方式中,由于第一存储设备及第二存储设备都备份了对端的数据,所以在收到访问对端数据的访问请求时,可以从本端备份的对端的备份数据中获取需要访问的数据,而无需从对端获取所访问的数据,从而提高了数据的访问效率。

[0154] 当所述Active-Active双活存储系统中的一个存储设备故障后,则可以通过备份数据接管故障存储设备的业务。如图10所示,在第一存储设备及第二存储设备之间的链路断开,或者第二存储设备故障后,可以通过第一存储设备中存储的第二存储设备的备份数据接管第二存储设备的业务。下面以在第一存储设备及第二存储设备之间的链路断开为例进行说明。具体如图11所示的流程图所示。

[0155] 步骤S111,第一存储设备和第二存储设备同时侦测对端的心跳。

[0156] 步骤S112,在侦测不到对端心跳时,第一存储设备和第二存储设备挂起正在执行的业务。

[0157] 挂起业务,即停止正在执行的访问请求。

[0158] 步骤S113,第一存储设备和第二存储设备修改全局视图及文件系统。

[0159] 当侦测不到对端心跳时,则第一存储设备和第二存储设备就要为接管对端的业务做准备,则会对全局视图及文件系统进行修改,会将全局视图中对端的物理节点对应的虚拟节点从全局视图中删除,将备份策略中对端的备份节点删除。例如第一存储设备将全局视图修改为(Vnode A,Vnode B),第二存储设备将全局视图修改为(Vnode C,Vnode D)。另外将文件系统中的shard视图中对端节点对应的虚拟节点的shard修改为本端节点对应的虚拟节点对应的shard。例如,将第一存储设备中的shard视图修改为Vnode A[0,2047],Vnode B[2048,4095];将第二存储设备中的shard视图修改为Vnode C[0,2047],Vnode D[2048,4095];同时将刷盘策略对端节点的卷删除。

[0160] 步骤S114,第一存储设备及第二存储设备都向仲裁设备发送仲裁请求。

[0161] 步骤S115,仲裁设备仲裁第一存储设备接管业务。

[0162] 仲裁设备可根据收到仲裁请求的先后顺序确定接管业务的设备,例如,先收到的仲裁请求对应的存储设备作为接管业务的设备。

[0163] 步骤S116,仲裁设备将仲裁结果分别通知到第一存储设备和第二存储设备。

[0164] 步骤S117,第二存储设备接收到所述通知后,解除与第二客户端的连接,即停止业务的执行。

[0165] 步骤S118,第一存储设备接收到所述通知后,将第二存储阵列的IP地址漂移至第一存储设备,并建立与第二客户端的连接。

[0166] 步骤S119,通过第二存储阵列的备份数据接管第二存储阵列的业务。

[0167] 由于所述第一存储设备中存储了所述第二存储阵列的备份数据,所以在第一存储

设备接收到所述第一存储设备中的数据的访问时，可以通过访问请求中的shard ID将第一客户端及第二客户端对第二存储设备中的数据的访问请求定位到对所述备份数据的访问，从而使第一客户端及第二客户端感知不到链路中断。

[0168] 在数据访问的过程中，由于备份策略及刷盘策略的更改，写入的数据则只会写入第一存储设备的节点中的内存，并且只在第一存储设备的卷中存储。

[0169] 以上对本申请实施例所提供方案进行了描述，本文中应用了具体个例对本申请的原理及实施方式进行了阐述，以上实施例的说明只是用于帮助理解本申请的方法及其核心思想；同时，对于本领域的一般技术人员，依据本申请的思想，在具体实施方式及应用范围上均会有改变之处，综上所述，本说明书内容不应理解为对本申请的限制。

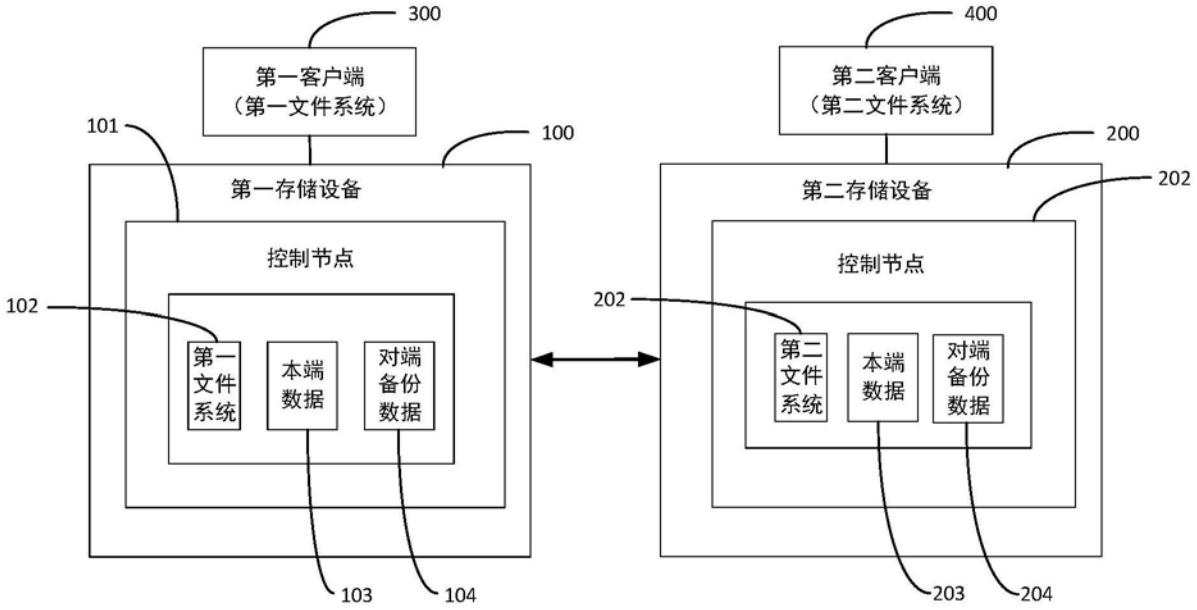

300
第一客户端
（第一文件系统）
101
100
第一存储设备
控制节点
102
第一文件系统
本端数据
对端备份数据
103
104
400
第二客户端
（第二文件系统）
200
202
第二存储设备
控制节点
202
第二文件系统
本端数据
对端备份数据
203
204

图1

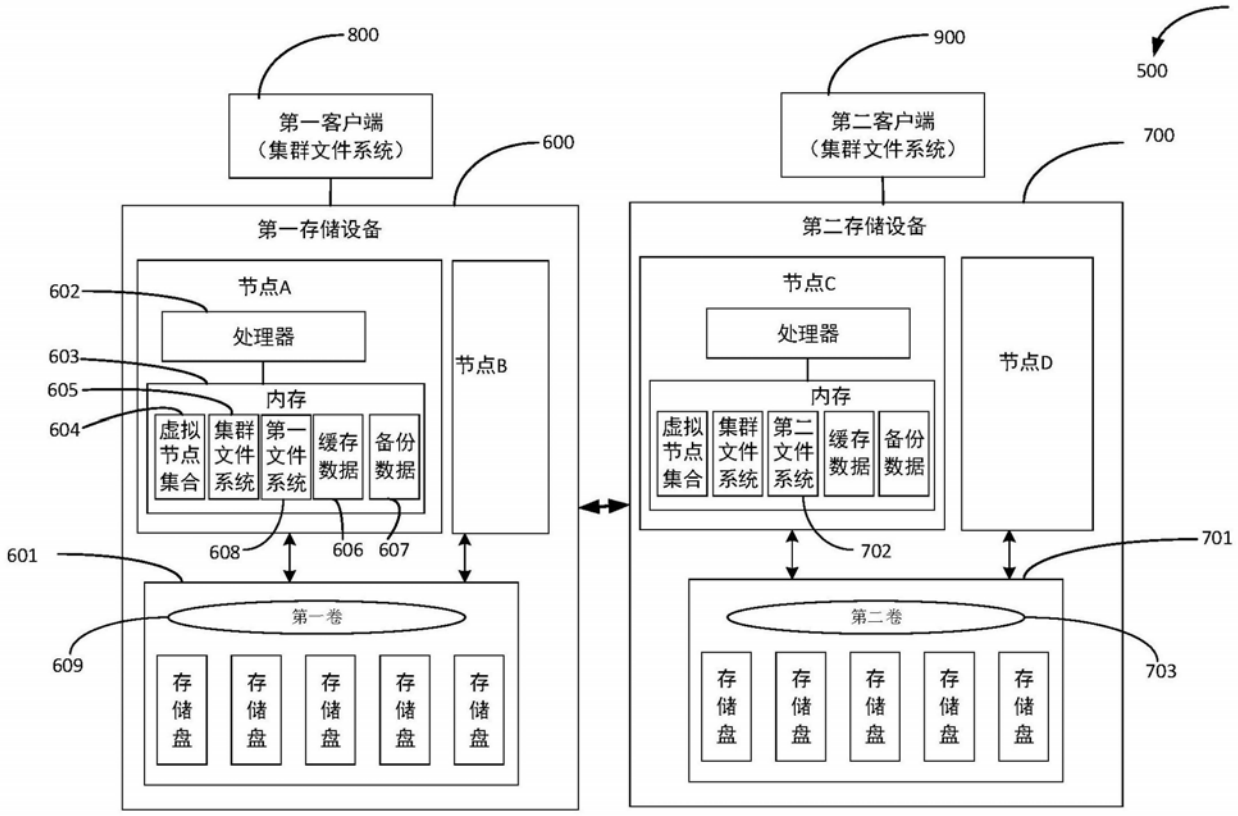

800
第一客户端
（集群文件系统）
600
第一存储设备
节点A
602
处理器
603
605
604
内存
虚拟节点集合
集群文件系统
第一文件系统
缓存数据
备份数据
节点B
608
606
607
601
第一卷
609
存储盘
900
500
第二客户端
（集群文件系统）
700
第二存储设备
节点C
处理器
内存
虚拟节点集合
集群文件系统
第二文件系统
缓存数据
备份数据
节点D
702
701
第二卷
703
存储盘

图2

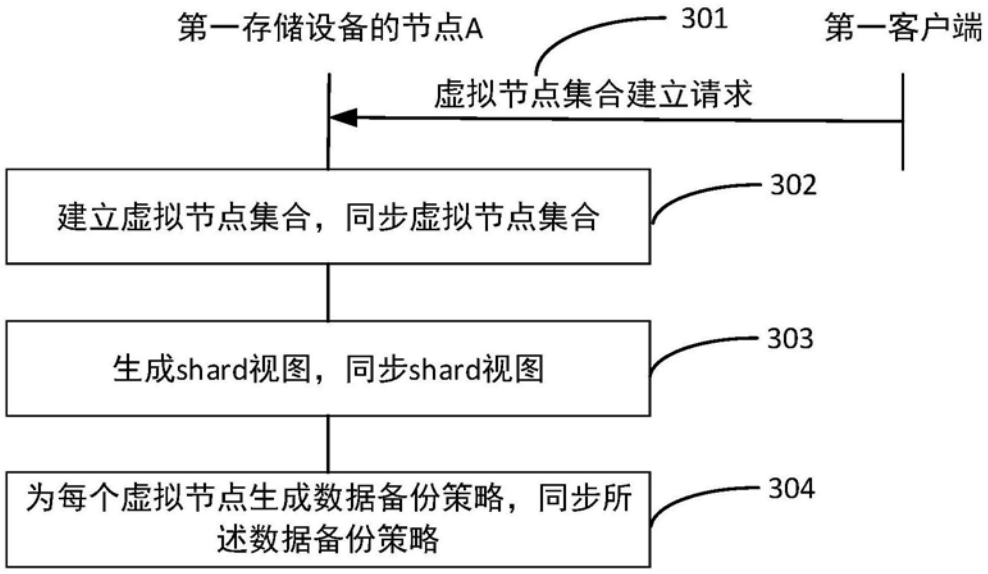

第一存储设备的节点A
301
第一客户端
虚拟节点集合建立请求
建立虚拟节点集合，同步虚拟节点集合
302
生成shard视图，同步shard视图
303
为每个虚拟节点生成数据备份策略，同步所述数据备份策略
304

图3A

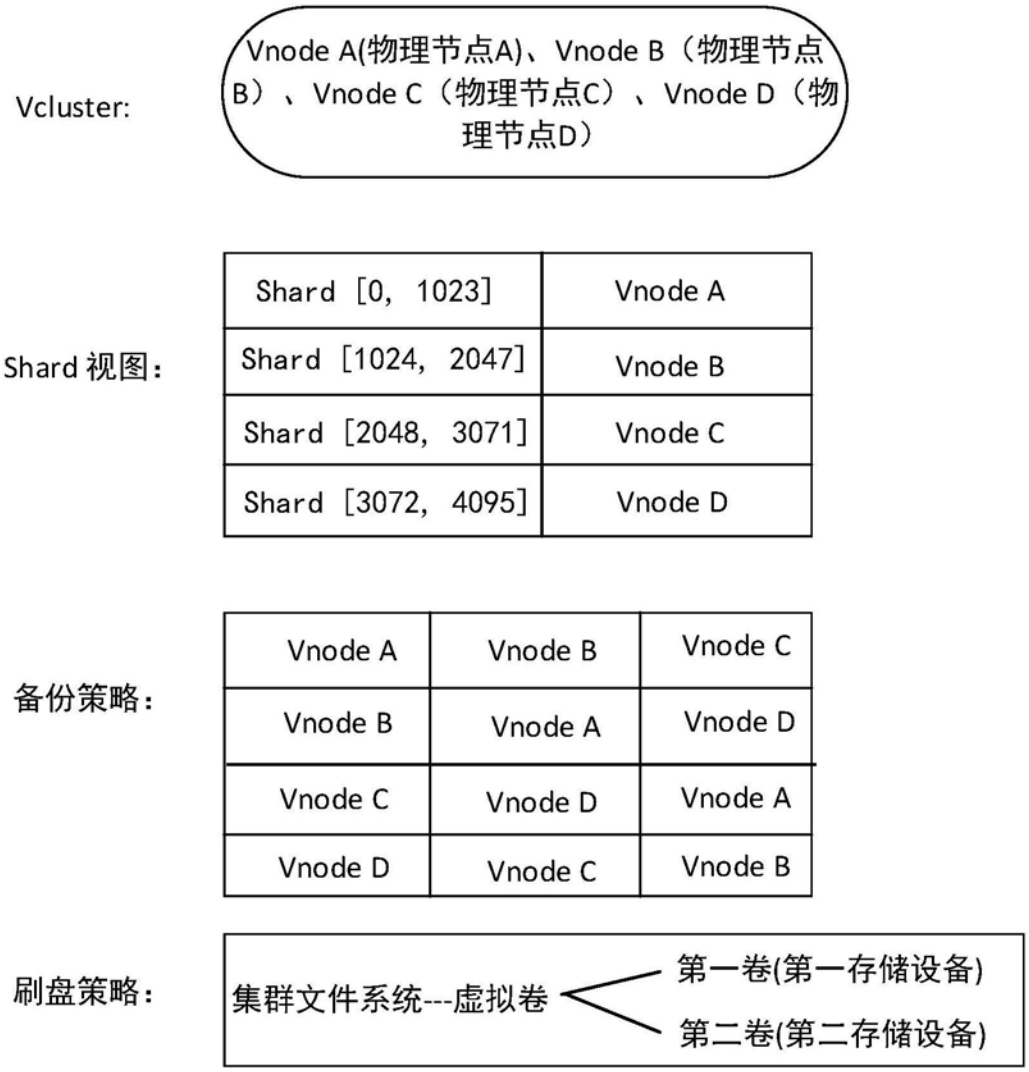

Vcluster:
Vnode A(物理节点A)、Vnode B（物理节点B）、Vnode C（物理节点C）、Vnode D（物理节点D）
Shard 视图：
Shard [0, 1023] Vnode A
Shard [1024, 2047] Vnode B
Shard [2048, 3071] Vnode C
Shard [3072, 4095] Vnode D
备份策略：
Vnode A Vnode B Vnode C
Vnode B Vnode A Vnode D
Vnode C Vnode D Vnode A
Vnode D Vnode C Vnode B
刷盘策略：
集群文件系统---虚拟卷
第一卷(第一存储设备)
第二卷(第二存储设备)

图3B

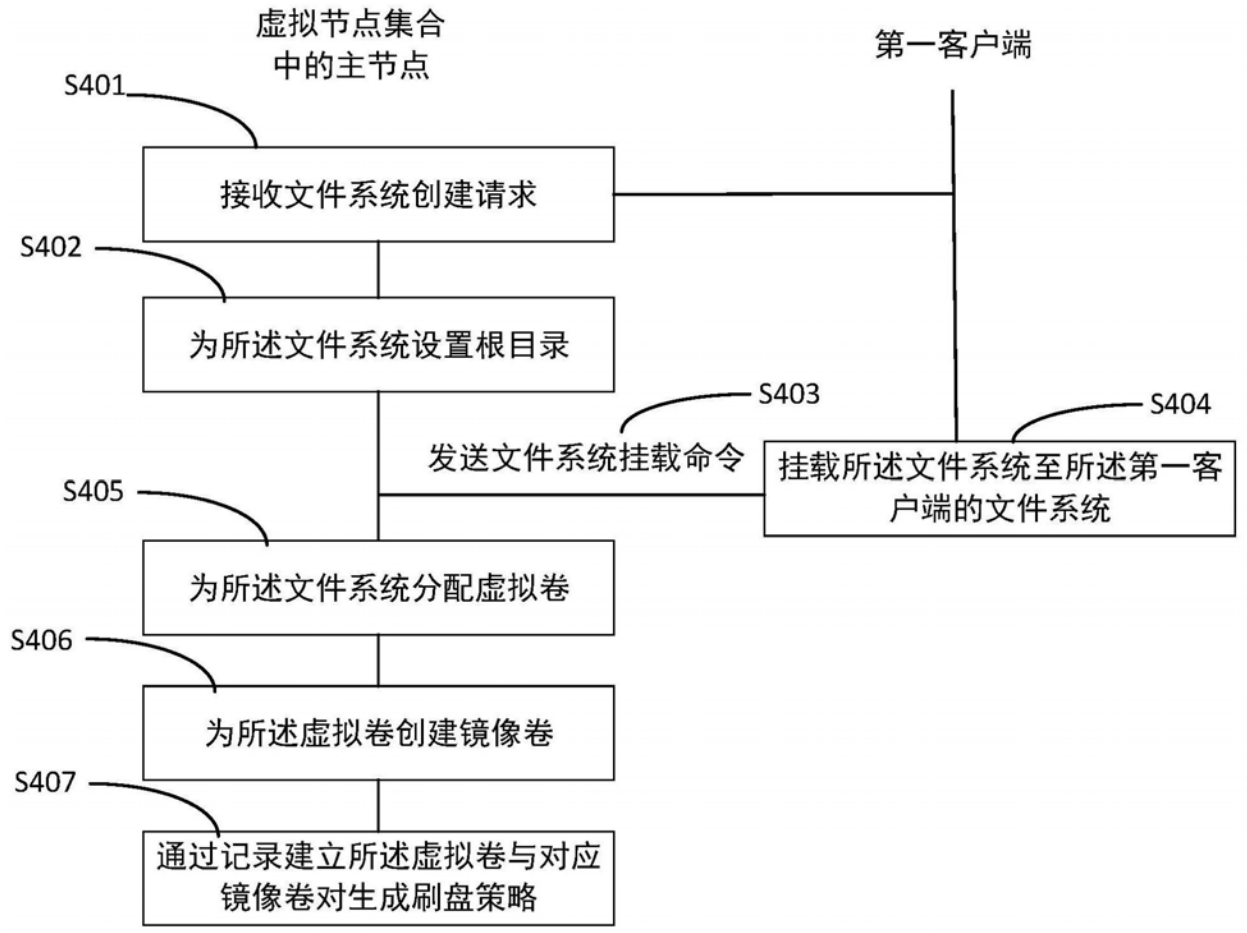

虚拟节点集合
中的主节点
第一客户端
S401
接收文件系统创建请求
S402
为所述文件系统设置根目录
S403
发送文件系统挂载命令
S404
挂载所述文件系统至所述第一客户端的文件系统
S405
为所述文件系统分配虚拟卷
S406
为所述虚拟卷创建镜像卷
S407
通过记录建立所述虚拟卷与对应镜像卷对生成刷盘策略

图4A

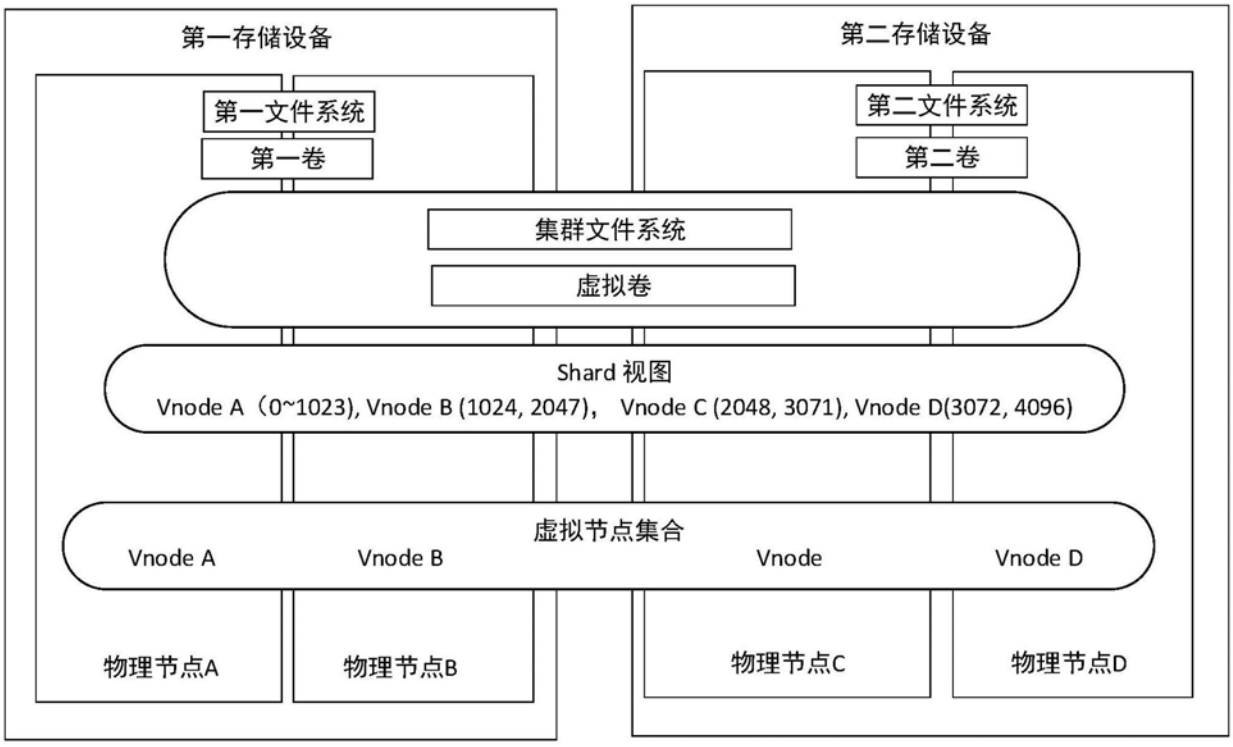

第一存储设备
第一文件系统
第一卷
第二存储设备
第二文件系统
第二卷
集群文件系统
虚拟卷
Shard 视图
Vnode A（0~1023), Vnode B (1024, 2047)， Vnode C (2048, 3071), Vnode D(3072, 4096)
虚拟节点集合
Vnode A
Vnode B
Vnode
Vnode D
物理节点A
物理节点B
物理节点C
物理节点D

图4B

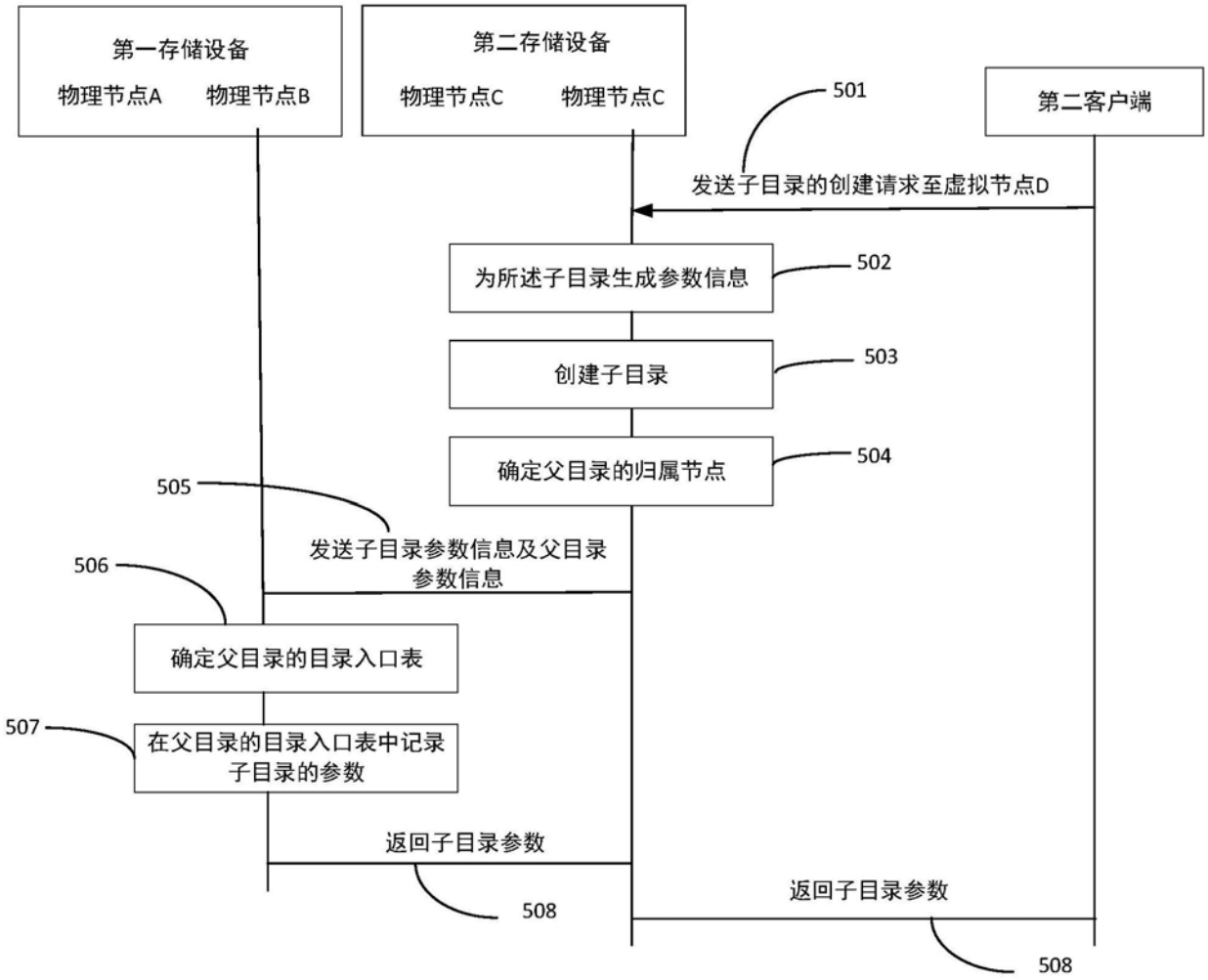

第一存储设备
物理节点A
物理节点B
第二存储设备
物理节点C
物理节点C
第二客户端
501
发送子目录的创建请求至虚拟节点D
为所述子目录生成参数信息
502
创建子目录
503
确定父目录的归属节点
504
505
发送子目录参数信息及父目录参数信息
506
确定父目录的目录入口表
507
在父目录的目录入口表中记录子目录的参数
返回子目录参数
508
返回子目录参数
508

图5

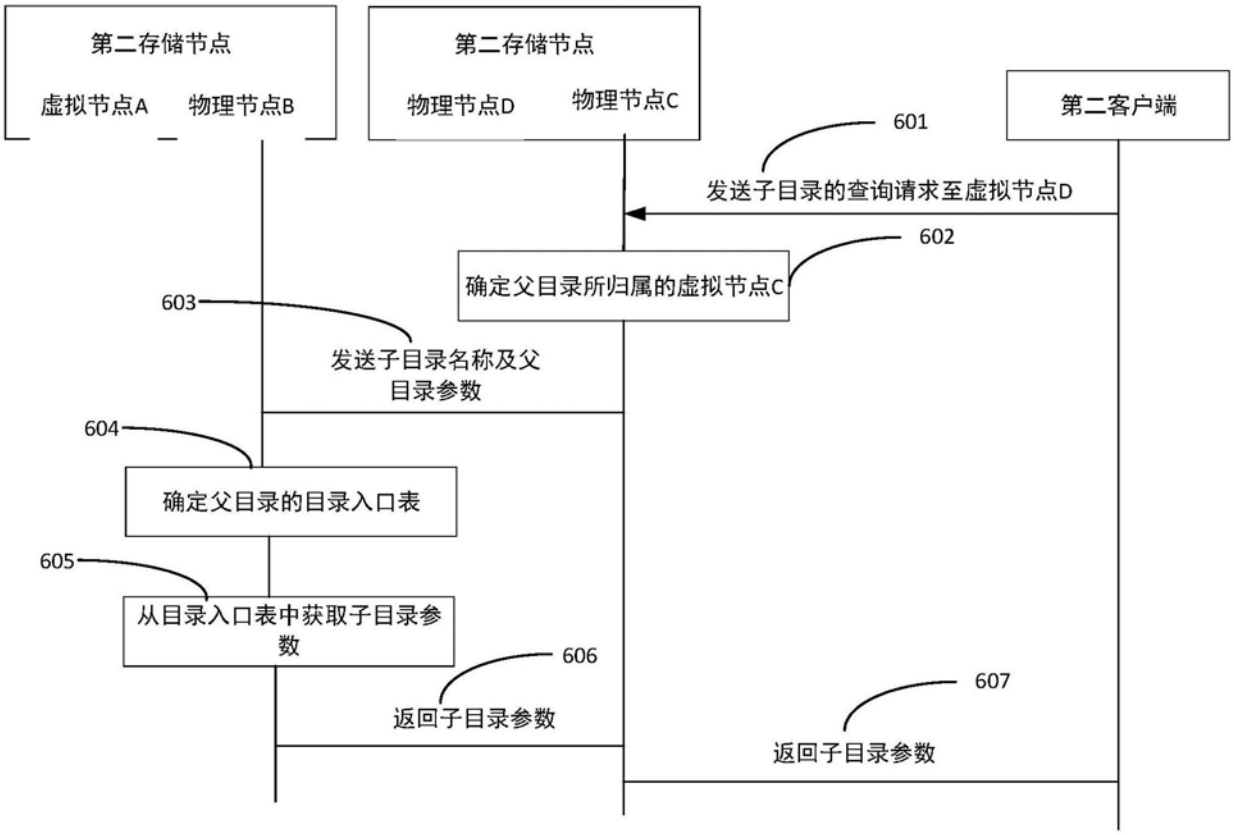

第二存储节点
虚拟节点A
物理节点B
第二存储节点
物理节点D
物理节点C
第二客户端
601
发送子目录的查询请求至虚拟节点D
602
确定父目录所归属的虚拟节点C
603
发送子目录名称及父目录参数
604
确定父目录的目录入口表
605
从目录入口表中获取子目录参数
606
返回子目录参数
607
返回子目录参数

图6

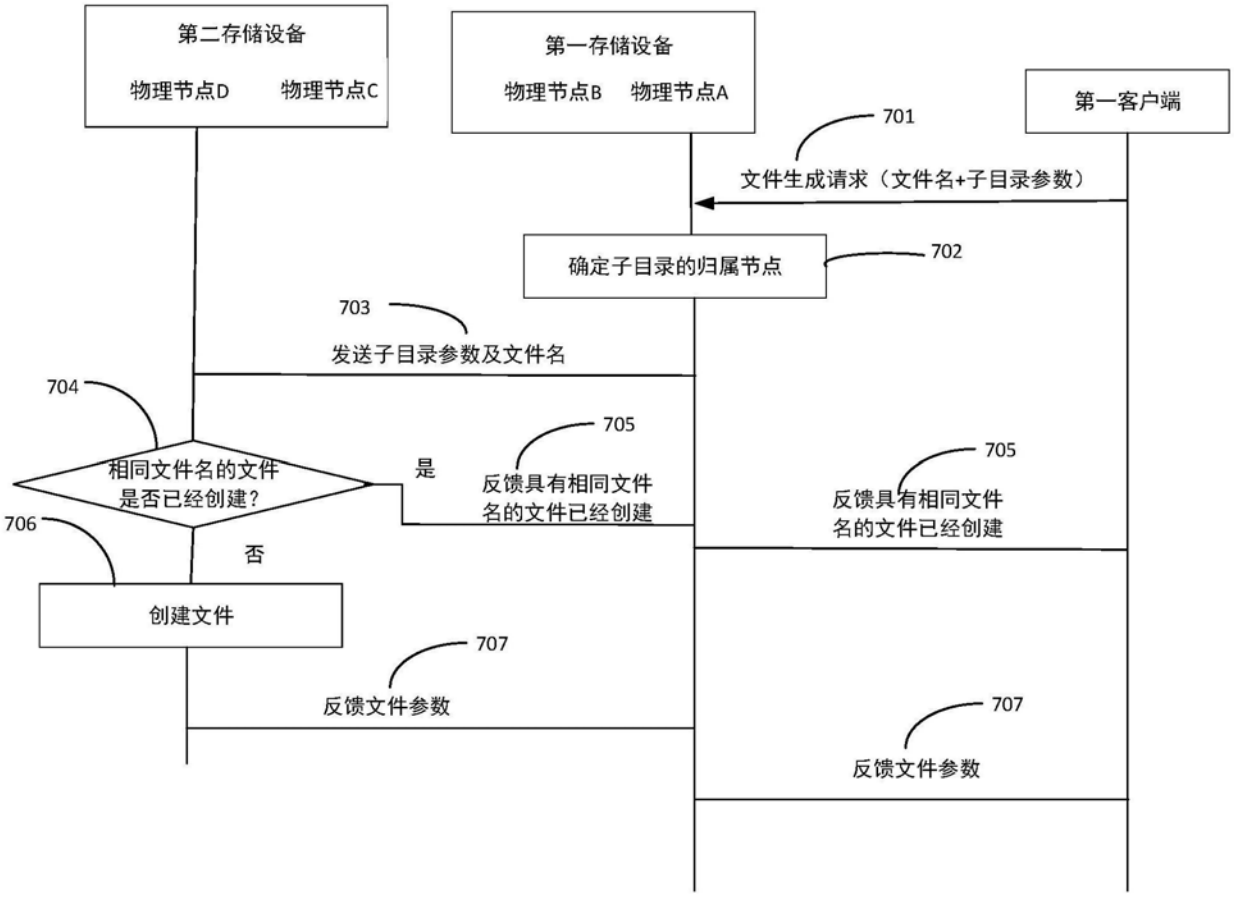

第二存储设备
物理节点D 物理节点C
第一存储设备
物理节点B 物理节点A
第一客户端
701
文件生成请求（文件名+子目录参数）
确定子目录的归属节点
702
703
发送子目录参数及文件名
704
相同文件名的文件
是否已经创建？
是
705
反馈具有相同文件
名的文件已经创建
705
反馈具有相同文件
名的文件已经创建
706
否
创建文件
707
反馈文件参数
707
反馈文件参数

图7

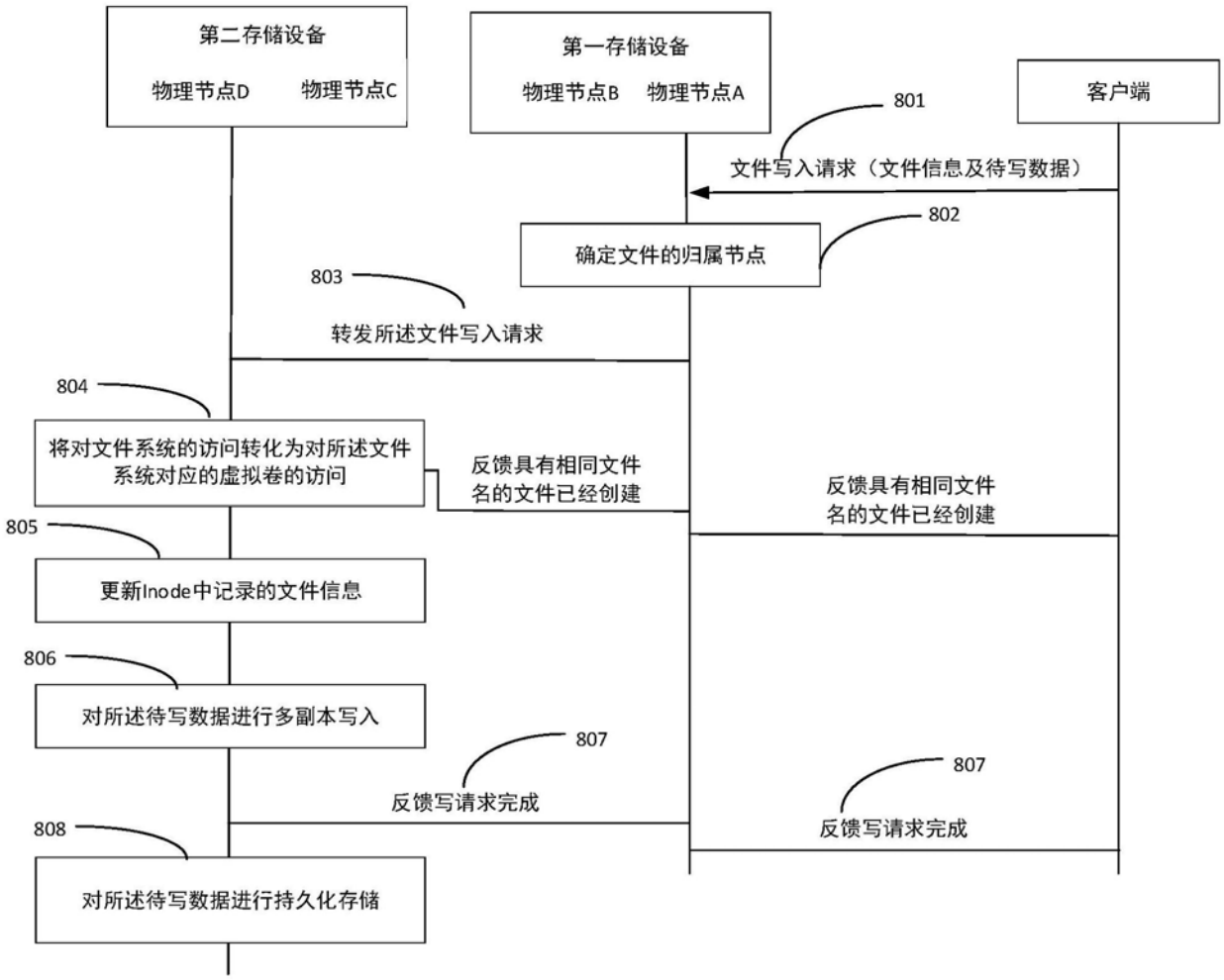

第二存储设备
物理节点D
物理节点C
第一存储设备
物理节点B
物理节点A
客户端
801
文件写入请求（文件信息及待写数据）
802
确定文件的归属节点
803
转发所述文件写入请求
804
将对文件系统的访问转化为对所述文件系统对应的虚拟卷的访问
反馈具有相同文件名的文件已经创建
反馈具有相同文件名的文件已经创建
805
更新Inode中记录的文件信息
806
对所述待写数据进行多副本写入
807
反馈写请求完成
807
反馈写请求完成
808
对所述待写数据进行持久化存储

图8

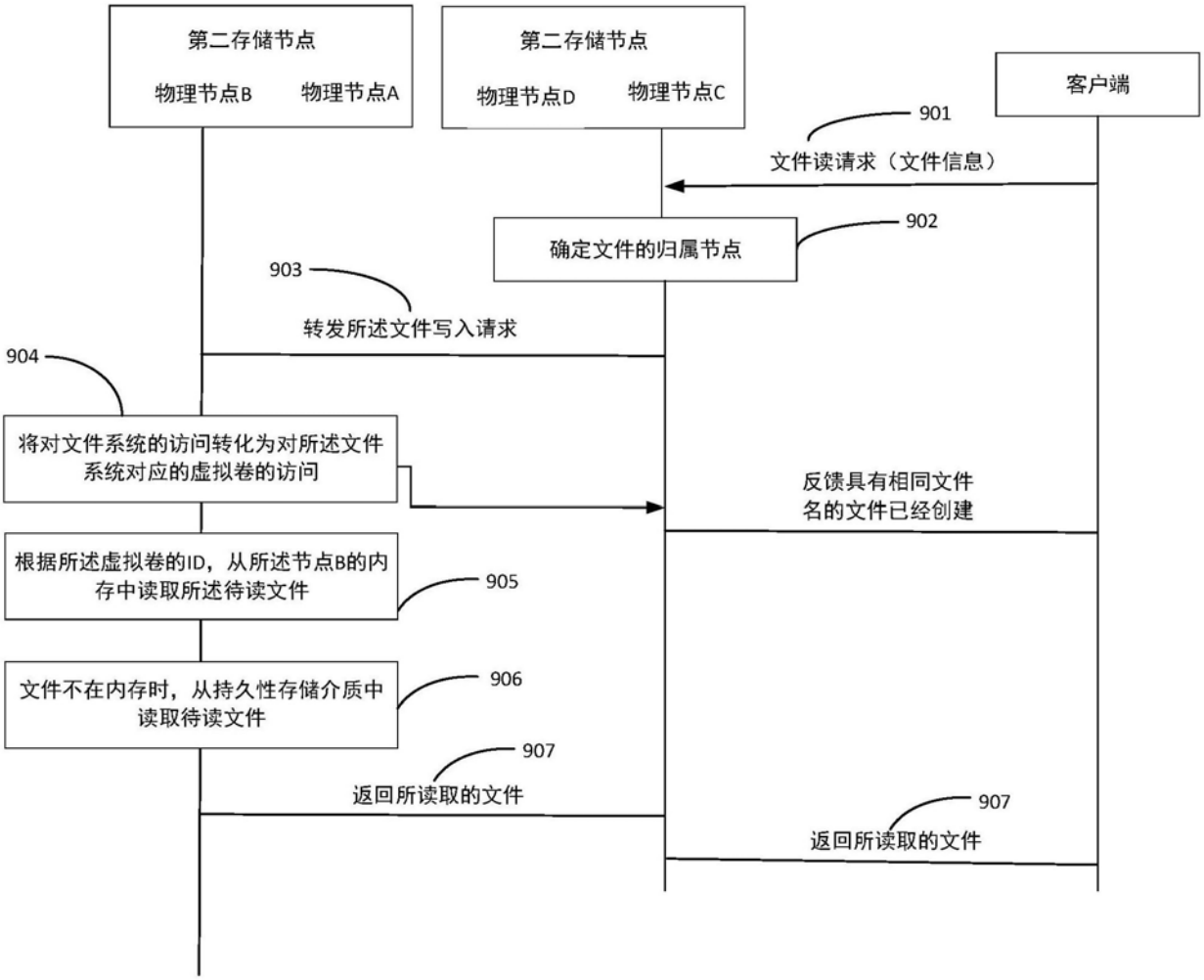

第二存储节点
物理节点B
物理节点A
第二存储节点
物理节点D
物理节点C
客户端
901
文件读请求（文件信息）
确定文件的归属节点
902
903
转发所述文件写入请求
904
将对文件系统的访问转化为对所述文件系统对应的虚拟卷的访问
反馈具有相同文件名的文件已经创建
根据所述虚拟卷的ID，从所述节点B的内存中读取所述待读文件
905
文件不在内存时，从持久性存储介质中读取待读文件
906
907
返回所读取的文件
907
返回所读取的文件

图9

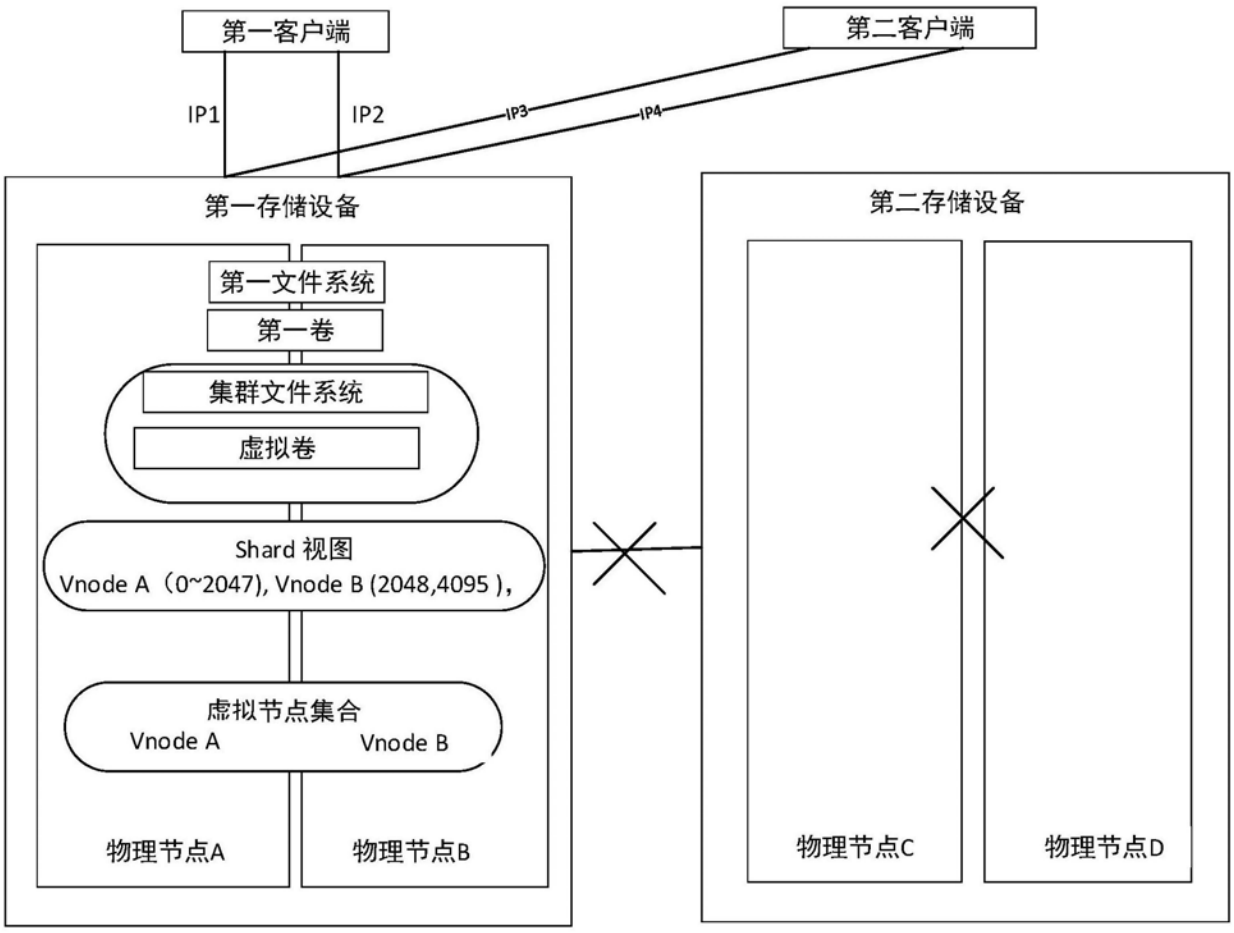

第一客户端
第二客户端
IP1
IP2
IP3
IP4
第一存储设备
第二存储设备
第一文件系统
第一卷
集群文件系统
虚拟卷
Shard 视图
Vnode A（0~2047), Vnode B (2048,4095 ),
虚拟节点集合
Vnode A
Vnode B
物理节点A
物理节点B
物理节点C
物理节点D

图10

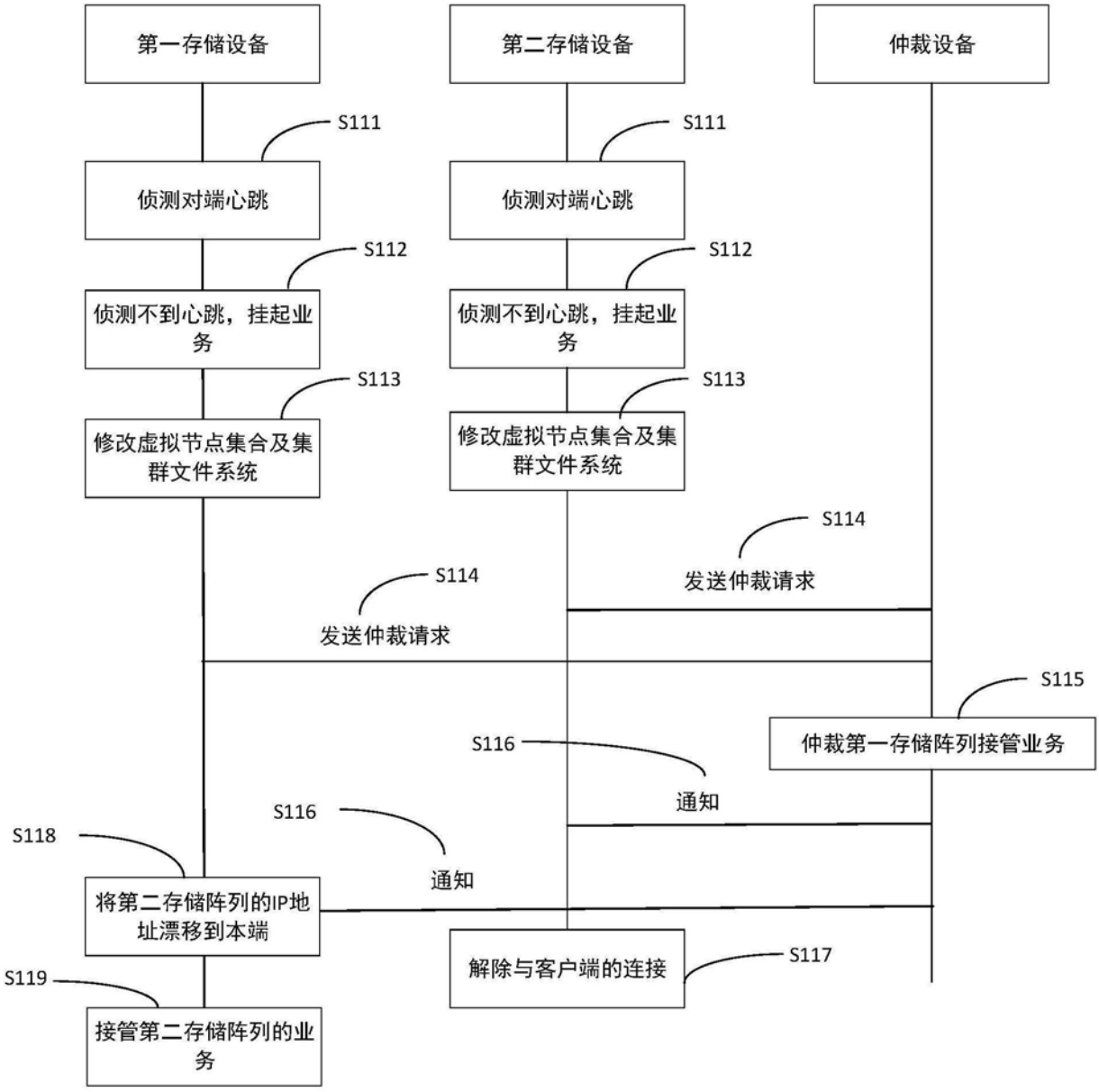

第一存储设备
第二存储设备
仲裁设备
S111
侦测对端心跳
S111
侦测对端心跳
S112
侦测不到心跳，挂起业务
S112
侦测不到心跳，挂起业务
S113
修改虚拟节点集合及集群文件系统
S113
修改虚拟节点集合及集群文件系统
S114
发送仲裁请求
S114
发送仲裁请求
S115
仲裁第一存储阵列接管业务
S116
通知
S116
通知
S118
将第二存储阵列的IP地址漂移到本端
解除与客户端的连接
S117
S119
接管第二存储阵列的业务

图11