

(12) 发明专利

(10) 授权公告号 CN 101414277 B

(45) 授权公告日 2010.06.09

(21) 申请号 200810225919.5

CN 1746855 A, 2006.03.15, 全文.

(22) 申请日 2008.11.06

CN 1617600 A, 2005.05.18, 全文.

(73) 专利权人 清华大学

CN 1529426 A, 2004.09.15, 全文.

地址 100084 北京市海淀区清华园北京
100084-82 信箱

审查员 何俊

(72) 发明人 郑纬民 余宏亮 向小佳

(74) 专利代理机构 北京路浩知识产权代理有限公司 11002

代理人 张国良

(51) Int. Cl.

G06F 11/14 (2006.01)

H04L 29/08 (2006.01)

(56) 对比文件

US 20030187847 A1, 2003.10.02, 全文.

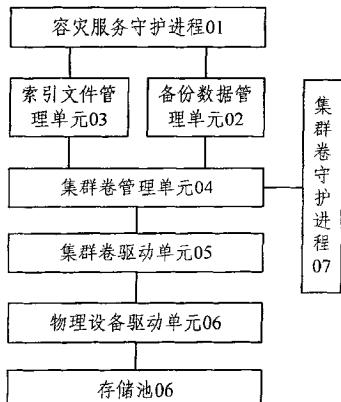
权利要求书 3 页 说明书 11 页 附图 4 页

(54) 发明名称

一种基于虚拟机的按需增量恢复容灾系统及方法

(57) 摘要

本发明涉及一种基于虚拟机的按需增量恢复容灾系统及方法，通过集成虚拟机进程上下文环境，能够构建应用相关的进程树，并支持对其进行冻结和截取一致的内存映像，在此基础上，提出了按需增量恢复方法，在通过内核拦截单元对虚拟机进程要访问文件系统数据进行监测的同时，在后台进行应用程序的相关数据块恢复，使虚拟机中的应用和恢复流程同时运行；恢复过程分为两阶段，即先恢复内存映像，再恢复文件系统或卷数据，通过该流程应用程序能够精确恢复到备份时的运行状态；该方法支持对客户端的多个备份任务和恢复任务的调度，支持对恢复数据的预取。本发明具有恢复时间短，能提高被保护应用的可靠性，对用户透明且成本低廉等优点，具有实用价值。



1. 一种基于虚拟机的按需增量恢复容灾系统，其特征在于，该系统包括相互连接的容灾中心端和客户端，所述容灾中心端由多台服务器构成存储共享集群，所述容灾中心端包括：

容灾服务守护单元，负责监听端口，接受来自客户端的备份或恢复请求，相应地进行数据的备份或恢复；

备份数据管理单元，用于对客户端的备份请求生成相应的内存映像卷和数据映像卷并存储；

集群卷管理单元，虚拟出共享存储池供所有节点使用，以逻辑卷的方式在存储池中存放和管理内存映像卷和数据映像卷；

索引文件管理单元，用于对存储池中的内存映像卷和数据映像卷进行检索和定位；

所述客户端包括：

虚拟机环境构建单元，用于在客户端宿主系统上建立独立的进程组与会话，封装用户需要保护的应用及相关资源，进而构建出客户端进程所处的虚拟机环境；

内核拦截单元，用于监测虚拟机环境的每次读写请求，查询和维护记录了数据恢复情况的恢复数据位图；

虚拟机冻结单元，用于在内核拦截单元监测到虚拟机环境中的进程有读写请求及查询恢复数据位图，若要读写的数据未恢复则在内核态发起虚拟机环境冻结操作；

虚拟机解冻单元，用于在要读写的数据写入客户端本地存储介质后，发起解冻虚拟机的操作，使虚拟机环境中的各个进程恢复运作；

请求转发守护单元，用于与容灾中心端通信，发出备份或恢复请求并传送、接收数据，集中完成数据请求管理、调度实现容灾恢复。

2. 如权利要求 1 所述的基于虚拟机的按需增量恢复容灾系统，其特征在于，该系统中客户端请求转发守护单元包括备份子单元和恢复子单元，其中：

所述备份子单元，用于接收用户发来的备份请求，并生成备份任务，根据备份任务的需求与容灾中心端建立连接，并转发该备份任务的后续请求到服务器端，同时根据各个备份任务的优先级集中对备份请求进行调度和流量控制；

所述恢复子单元，用于接收用户以及内核拦截单元发来的恢复请求，并生成恢复任务，根据恢复任务的需求与容灾中心端建立连接，记录容灾中心端传送回来的数据，写入本地存储介质，同时对来自内核拦截单元的恢复请求，还负责回调函数的执行而完成数据恢复，并根据各个恢复任务的优先级集中对恢复请求进行调度和流量控制。

3. 如权利要求 1 所述的基于虚拟机的按需增量恢复容灾系统，其特征在于，该系统中所述容灾中心端还包括集群卷守护单元，与所述集群卷管理单元通信，用于维护存储共享集群的状态，保证共享集群中的有效节点数，同步对存储池所存储内存映像卷和数据映像卷所作的修改。

4. 如权利要求 1 所述的基于虚拟机的按需增量恢复容灾系统，其特征在于，该系统中所述容灾中心端还包括集群卷驱动单元和物理设备驱动单元，所述集群卷驱动单元用于维护内存映像卷和数据映像卷的逻辑数据块与物理数据块映射信息，各逻辑卷读写命令的调度和派发，所述集群卷驱动单元下层是物理设备驱动单元，所述物理设备驱动单元通过存储区域网络连接所述共享存储池。

5. 一种利用权利要求 1 所述系统的基于虚拟机的按需增量恢复容灾方法, 其特征在于, 该方法包括以下步骤 :

s1. 0, 在客户端上构建虚拟机环境, 将要保护的应用包在虚拟机环境中;

s1. 1, 根据客户端的备份请求在容灾中心端生成虚拟机环境在指定时间点的内存映像卷和数据映像卷, 完成数据备份;

s1. 2, 客户端启动恢复流程, 根据用户发起的恢复请求生成恢复任务, 并向容灾中心端发送恢复请求, 同时通知内核拦截单元初始化恢复数据位图;

s1. 3, 容灾中心端根据恢复请求检索和定位虚拟机环境在指定时间点的内存映像卷, 并将其发送到客户端;

s1. 4, 客户端根据接收内存映像卷获取虚拟机环境在指定时间点的内存映像, 进行内存映像重构而恢复虚拟机环境在指定时间点的正常运行状态, 并行执行步骤 s1. 5 和 s1. 5' ;

s1. 5, 在客户端开启内核通用块设备的读写请求拦截功能, 当虚拟机环境中的进程访问文件系统时首先查询恢复数据位图, 若要读写的数据已恢复, 则直接访问, 否则执行步骤 s1. 6 ;

s1. 6, 客户端发起虚拟机环境的冻结操作, 由内核拦截单元发出读写数据的恢复请求;

s1. 7, 将步骤 s1. 6 中的恢复请求挂载到步骤 s1. 2 所建立恢复任务的队列中, 并赋予最高优先级, 等待容灾中心端的响应;

s1. 8, 客户端收到来自容灾中心端的所述读写数据, 更新恢复数据位图, 标志该读写数据为已恢复, 并将其写入本地存储介质;

s1. 9, 解冻虚拟机, 使虚拟机环境中的各个进程恢复运作;

s1. 5', 客户端根据由步骤 s1. 2 所建立的恢复任务, 在后台不断发送数据块恢复请求到容灾中心端进行数据块恢复, 在恢复过程中通过查询恢复数据位图避免数据重复恢复。

6. 如权利要求 5 所述的基于虚拟机的按需增量恢复容灾方法, 其特征在于, 该方法中步骤 s1. 4 中进行内存映像重构恢复虚拟机环境在指定时间点的正常运行状态包括步骤 :

s1. 4a, 根据内存映像中的进程树派生一个内核进程, 修改内核进程内核栈, 令内核进程在系统调用退出时切换为缺省的自恢复进程;

s1. 4b, 所述内核进程根据内存映像中的进程树派生出其子进程, 所述子进程为自恢复进程, 再次修改内核栈, 构造第二次系统调用退出, 进而切换为指定时间点时的用户态根进程;

s1. 4c, 根据内存映像中的进程树结构按从根到叶的顺序依次派生出新的子进程, 直到内存映像中的所有进程都被派生出来为止, 设置各个所派生出的进程为深度睡眠状态;

s1. 4d, 在客户端另启动一个独立进程, 该进程位于虚拟机环境外, 该独立进程根据用户的指令将虚拟机环境中所有进程的状态按照内存映像中的记录进行重置, 从而恢复虚拟机环境在指定时间点的正常运行状态。

7. 如权利要求 5 所述的基于虚拟机的按需增量恢复容灾方法, 其特征在于, 该方法中步骤 s1. 0 构建虚拟机环境包括步骤 : :

s1. 0a, 扩充客户端操作系统内核中进程的结构, 使之增加代表进程所处的虚拟机进程

上下文环境的域；

s 1.0b, 开启一个用户态进程, 完成输入输出设备的切换, 进而通过系统调用切入内核态；

s1.0c, 将上述用户态进程切换到用户空间, 在虚拟机上下文环境中执行 /sbin/init, 使用用户态进程成为该虚拟机环境的根进程, 然后派生出相关子进程, 创建相关系统服务；

s1.0d, 建立本地操作系统中的进程上下文与虚拟机进程上下文的切换机制, 构建出虚拟机环境。

8. 如权利要求 5 所述的基于虚拟机的按需增量恢复容灾方法, 其特征在于, 在步骤 s1.6 中, 发出读写数据的恢复请求后, 选择预取数据进行预取, 预取方法为提交与读写数据物理位置邻近的各个数据块的恢复请求, 邻近位置由设定的地址偏移阈值决定。

9. 如权利要求 8 所述的基于虚拟机的按需增量恢复容灾方法, 其特征在于, 所述地址偏移阈值为动态调优参数, 调优的方法为 : 预取数据为虚拟机环境中的进程要访问文件所需数据时, 增加所述地址偏移阈值的绝对值, 反之则减小。

10. 如权利要求 8 所述的基于虚拟机的按需增量恢复容灾方法, 其特征在于, 所述预取数据恢复请求的优先级低于步骤 s1.6 中读写数据的恢复请求优先级, 高于步骤 s1.5' 中后台所发的数据块恢复请求优先级。

11. 如权利要求 5 所述的基于虚拟机的按需增量恢复容灾方法, 其特征在于, 在步骤 s1.5' 中, 后台按照预定的恢复策略不断发送数据块恢复请求到容灾中心端进行数据块恢复, 所述恢复策略为 : 按敏感度高低顺发送数据块恢复请求, 先发送敏感度高的数据块, 所述敏感度设置在数据块中, 通过统计每个数据块在一个固定大小时间窗口 w 内的读写次数获得。

12. 如权利要求 5 所述的基于虚拟机的按需增量恢复容灾方法, 其特征在于, 在步骤 s1.8 中, 对位图进行更新后还包括以游程码的压缩方式进行位图压缩步骤。

13. 如权利要求 5 所述的基于虚拟机的按需增量恢复容灾方法, 其特征在于, 步骤 s1.8 中, 对位图进行更新时采用读写锁机制, 在写操作更新数据的同时, 并不删除老数据, 直到对老数据正在进行的读访问全部结束为止, 该过程中新发生的读操作定位到新的数据。

14. 如权利要求 5 所述的基于虚拟机的按需增量恢复容灾方法, 其特征在于, 在步骤 s1.2 中, 客户端对于每个恢复请求, 将其通过指针链接成为一个双向链表, 并在内存中建立恢复请求 hash 表, 根据所赋予的请求号将各个恢复请求放置到各个哈希桶中。

15. 如权利要求 5 所述的基于虚拟机的按需增量恢复容灾方法, 其特征在于, 在步骤 s1.8 及步骤 s1.5' 中, 当客户端接收到来自容灾中心端所恢复的数据后, 先写入位图, 并告知后续程序数据已经传送到, 若数据没能成功写入, 接收到的数据会一直保留并被重写, 直到数据顺利写入。

一种基于虚拟机的按需增量恢复容灾系统及方法

技术领域

[0001] 本发明涉及操作系统和网络存储领域,具体涉及一种基于虚拟机的按需增量恢复容灾系统及方法。

背景技术

[0002] 容灾技术能够在各种自然灾害和人为破坏的情况下,保证数据的安全和关键业务的不间断运行,构建高可靠的系统;当前的容灾系统,按照数据复制运行的位置来分,包括如下几类:基于存储设备(Storage-Based)的容灾系统,该类系统实现于专用的物理存储设备之上,如运行于EMC Clariion阵列上的MirrorView和Symmetrix存储阵列上的SRDF、IBM公司的PPRC、日立公司的TrueCopy;基于主机操作系统软件(Host-Based)的容灾系统,例如Symantec公司的VeritasVolume Replicator(VVR),通过对存储卷组RVG的复制来达到容灾的目的,类似的产品还有微软的卷影复制等;基于存储交换机(SAN-Based)的容灾系统,代表有Cisco的SANTap,以及FalconStor的IPStor等,都是在存储交换设备一级实现数据的备份;基于数据库/软件应用的容灾系统,代表有Oracle的DataGuard,DB2的远程Q复制等,这些系统都采用了对数据逻辑操作的复制技术,通过扫描和记录数据库日志实现,节省成本,但对数据库和应用有强依赖性;综上所述,目前已有的系统还存在如下全部或部分缺点:无法保留应用的运行状态,只能做到数据级的复制;依赖专用设备,成本高且适用范围小;依赖专门的应用;数据拷贝、恢复速度慢,服务停止时间过长。

发明内容

[0003] 本发明的目的是提供一种基于虚拟机的按需增量恢复容灾系统及方法,不依赖物理设备,能够保护任意应用的快速容灾恢复系统。

[0004] 为实现上述目的,本发明采用如下技术方案:

[0005] 一种基于虚拟机的按需增量恢复容灾系统,该系统包括相互连接的容灾中心端和客户端,所述容灾中心端由多台服务器构成存储共享集群,所述容灾中心端包括:

[0006] 容灾服务守护单元,负责监听端口,接受来自客户端的备份或恢复请求,相应地进行数据的备份或恢复;

[0007] 备份数据管理单元,用于对客户端的备份请求生成相应的内存映像卷和数据映像卷并存储;

[0008] 集群卷管理单元,虚拟出共享存储池供所有节点使用,以逻辑卷的方式在存储池中存放和管理内存映像卷和数据映像卷;

[0009] 索引文件管理单元,用于对存储池中的内存映像卷和数据映像卷进行检索和定位;

[0010] 所述客户端包括:

[0011] 虚拟机环境构建单元,用于在客户端宿主系统上建立独立的进程组与会话,封装用户需要保护的应用及相关资源,进而构建出客户端进程所处的虚拟机环境;

[0012] 内核拦截单元,用于监测虚拟机环境的每次读写请求,查询和维护记录了数据恢复情况的恢复数据位图;

[0013] 虚拟机冻结单元,用于在内核拦截单元监测到虚拟机环境中的进程有读写请求及查询恢复数据位图,若要读写的数据未恢复则在内核态发起虚拟机环境冻结操作;

[0014] 虚拟机解冻单元,用于在要读写的数据写入客户端本地存储介质后,发起解冻虚拟机的操作,使虚拟机环境中的各个进程恢复运作;

[0015] 请求转发守护单元,用于与容灾中心端通信,发出备份或恢复请求并传送、接收数据,集中完成数据请求管理、调度实现容灾恢复。

[0016] 其中,该系统中客户端请求转发守护单元包括备份子单元和恢复子单元,其中:

[0017] 所述备份子单元,用于接收用户发来的备份请求,并生成备份任务,根据备份任务的需求与容灾中心端建立连接,并转发该备份任务的后续请求到服务器端,同时根据各个备份任务的优先级集中对备份请求进行调度和流量控制;

[0018] 所述恢复子单元,用于接收用户以及内核拦截单元发来的恢复请求,并生成恢复任务,根据恢复任务的需求与容灾中心端建立连接,记录容灾中心端传送回来的数据,写入本地存储介质,同时对来自内核拦截单元的恢复请求,还负责回调函数的执行而完成数据恢复,并根据各个恢复任务的优先级集中对恢复请求进行调度和流量控制。

[0019] 其中,该系统中所述容灾中心端还包括集群卷守护单元,与所述集群卷管理单元通信,用于维护存储共享集群的状态,保证共享集群中的有效节点数,同步对存储池所存储内存映像卷和数据映像卷所作的修改。

[0020] 其中,该系统中所述容灾中心端还包括集群卷驱动单元和物理设备驱动单元,所述集群卷驱动单元用于维护内存映像卷和数据映像卷的逻辑数据块与物理数据块映射信息,各逻辑卷读写命令的调度和派发,所述集群卷驱动单元下层是物理设备驱动单元,所述物理设备驱动单元通过存储区域网络连接所述共享存储池。

[0021] 本发明还提供了一种利用上述系统的基于虚拟机的按需增量恢复容灾方法,该方法包括以下步骤:

[0022] s1. 0,在客户端上构建虚拟机环境,将要保护的应用包容在虚拟机环境中;

[0023] s1. 1,根据客户端的备份请求在容灾中心端生成虚拟机环境在指定时间点的内存映像卷和数据映像卷,完成数据备份;

[0024] s1. 2,客户端启动恢复流程,根据用户发起的恢复请求生成恢复任务,并向容灾中心端发送恢复请求,同时通知内核拦截单元初始化恢复数据位图;

[0025] s1. 3,容灾中心端根据恢复请求检索和定位虚拟机环境在指定时间点的内存映像卷,并将其发送到客户端;

[0026] s1. 4,客户端根据接收内存映像卷获取虚拟机环境在指定时间点的内存映像,进行内存映像重构而恢复虚拟机环境在指定时间点的正常运行状态,并行执行步骤 s1. 5 和 s1. 5' ;

[0027] s1. 5,在客户端开启内核通用块设备的读写请求拦截功能,当虚拟机环境中的进程访问文件系统时首先查询恢复数据位图,若要读写的数据已恢复,则直接访问,否则执行步骤 s1. 6 ;

[0028] s1. 6,客户端发起虚拟机环境的冻结操作,由内核拦截单元发出读写数据的恢复

请求；

[0029] s1.7, 将步骤 s1.6 中的恢复请求挂载到步骤 s1.2 所建立恢复任务的队列中，并赋予最高优先级，等待容灾中心端的响应；

[0030] s1.8, 客户端收到来自容灾中心端的所述读写数据，更新恢复数据位图，标志该读写数据为已恢复，并将其写入本地存储介质；

[0031] s1.9, 解冻虚拟机，使虚拟机环境中的各个进程恢复运作；

[0032] s1.5'，客户端根据由步骤 s1.2 所建立的恢复任务，在后台不断发送数据块恢复请求到容灾中心端进行数据块恢复，在恢复过程中通过查询恢复数据位图避免数据重复恢复。

[0033] 其中，该方法中步骤 s1.4 中进行内存映像重构恢复虚拟机环境在指定时间点的正常运行状态包括步骤：

[0034] s1.4a, 根据内存映像中的进程树派生一个内核进程，修改内核进程内核栈，令内核进程在系统调用退出时切换为缺省的自恢复进程；

[0035] s1.4b, 所述内核进程根据内存映像中的进程树派生出其子进程，所述子进程为自恢复进程，再次修改内核栈，构造第二次系统调用退出，进而切换为指定时间点时的用户态根进程；

[0036] s1.4e, 根据内存映像中的进程树结构按从根到叶的顺序依次派生出新的子进程，直到内存映像中的所有进程都被派生出来为止，设置各个所派生出的进程为深度睡眠状态；

[0037] s1.4d, 在客户端另启动一个独立进程，该进程位于虚拟机环境外，该独立进程根据用户的指令将虚拟机环境中所有进程的状态按照内存映像中的记录进行重置，从而恢复虚拟机环境在指定时间点的正常运行状态。

[0038] 其中，该方法中步骤 s1.0 构建虚拟机环境包括步骤：：

[0039] s1.0a, 扩充客户端操作系统内核中进程的结构，使之增加代表进程所处的虚拟机进程上下文环境的域；

[0040] s1.0b, 开启一个用户态进程，完成输入输出设备的切换，进而通过系统调用切入内核态；

[0041] s1.0c, 将上述用户态进程切换到用户空间，在虚拟机上下文环境中执行 /sbin/init，使用户态进程成为该虚拟机环境的根进程，然后派生出相关子进程，创建相关系统服务；

[0042] s1.0d, 建立本地操作系统中的进程上下文与虚拟机进程上下文的切换机制，构建出虚拟机环境。

[0043] 其中，在步骤 s1.6 中，发出读写数据的恢复请求后，选择预取数据进行预取，预取方法为提交与读写数据物理位置邻近的各个数据块的恢复请求，邻近位置由设定的地址偏移阈值决定。

[0044] 其中，所述地址偏移阈值为动态调优参数，调优的方法为：预取数据为虚拟机环境中的进程要访问文件所需数据时，增加所述地址偏移阈值的绝对值，反之则减小。

[0045] 其中，所述预取数据恢复请求的优先级低于步骤 s1.6 中读写数据的恢复请求优先级，高于步骤 s1.5' 中后台所发的数据块恢复请求优先级。

[0046] 其中，在步骤 s1.5' 中，后台按照预定的恢复策略不断发送数据块恢复请求到容灾中心端进行数据块恢复，所述恢复策略为：按敏感度高低顺发送数据块恢复请求，先发送敏感度高的数据块，所述敏感度设置在数据块中，通过统计每个数据块在一个固定大小时间窗口 w 内的读写次数获得。

[0047] 其中，在步骤 s1.8 中，对位图进行更新后还包括以游程码的压缩方式进行位图压缩步骤。

[0048] 其中，步骤 s1.8 中，对位图进行更新时采用读写锁机制，在写操作更新数据的同时，并不删除老数据，直到对老数据正在进行的读访问全部结束为止，该过程中新发生的读操作定位到新的数据。

[0049] 其中，在步骤 s1.2 中，客户端对于每个恢复请求，将其通过指针链接成为一个双向链表，并在内存中建立恢复请求 hash 表，根据所赋予的请求号将各个恢复请求放置到各个哈希桶中。

[0050] 其中，在步骤 s1.8 及步骤 s1.5' 中，当客户端接收到来自容灾中心端所恢复的数据后，先写入位图，并告知后续程序数据已经传送到，若数据没能成功写入，接收到的数据会一直保留并被重写，直到数据顺利写入。

[0051] 利用本发明提供的基于虚拟机的按需增量恢复容灾系统及方法，具有以下有益效果：

[0052] 1) 恢复与程序运行的并行，能够针对应用优先获取关键数据集，并在后台按照一定策略执行完全数据恢复，减少了应用与服务终止的时间；

[0053] 2) 能够保留应用的全部内存映像，同时保证一致性，能够将应用和服务恢复到备份时的运行状态；

[0054] 3) 通过构建一个独立的虚拟机进程上下文环境，减少环境内进程与外部进程间的依赖，能够保证恢复的正确性和独立性；

[0055] 4) 作为代理的请求转发守护进程，能够集中管理客户端的备份与恢复任务，控制网络流量，对数据块请求安装优先级进行调度；

[0056] 5) 服务器端能够支持全量和增量备份等不同模式，通过建立索引加强对备份数据的管理，其实现对恢复应用透明，仅提供符合 Posix 语义的统一读写接口。

附图说明

[0057] 图 1 为本发明基于虚拟机的按需增量恢复容灾系统结构示意图；

[0058] 图 2 为本实施例中容灾中心端的结构图；

[0059] 图 3 为本实施例中客户端的结构图；

[0060] 图 4 为本实施例基于虚拟机的按需增量恢复容灾方法示意图；

[0061] 图 5 为本实施例基于虚拟机的按需增量恢复容灾方法流程图。

[0062] 图中：1、客户端；2、服务器集群；3、SAN 共享存储目标器；4、广域网；5、光纤 / 以太网交换机；6、物理存储设备。

具体实施方式

[0063] 下面结合附图说明基于虚拟机的按需增量恢复容灾系统及方法。

[0064] 本发明提供的基于虚拟机的按需增量恢复容灾系统及方法,首先,不依赖于具体的设备和应用;其次,集成了客户端虚拟机进程上下文环境,能够构建应用相关的进程树,并支持对其进行冻结和截取一致的内存映像;在此基础上实现的按需增量恢复分为两阶段,即先恢复内存映像,再恢复文件系统或卷数据,能够恢复应用程序到备份时的运行状态,同时支持恢复和应用的同时运行,缩短服务停止时间;最后,支持对客户端的多个备份任务和恢复任务的调度,支持对恢复数据的预取。本发明具有恢复时间短,能提高被保护应用的可靠性,纯软件实现,对用户透明且成本低廉等优点,具有实用价值。

[0065] 本实施例中基于虚拟机的按需增量恢复容灾系统,其主要构成如图1,其构成包括申请容灾服务的客户端1和提供容灾服务的容灾中心端,容灾中心端由共享SAN(Storage Area Network)共享存储目标器3的服务器集群2构成,其中客户端1通过广域网4与服务器集群2连接,服务器集群2通过光纤/以太网交换机5与SAN共享存储目标器3连接,SAN共享存储目标器3连接物理存储设备6,每个服务器节点都能够访问整个存储空间,提供相同的服务,一个服务器节点故障可由其他节点或新加入节点替代,具有好的可用性。上述SAN网络共享存储目标器3是构成共享存储池的基本单元,通过其上运行的目标器软件,将物理存储设备6共享到网络上,物理存储设备6头部保存着虚拟存储池元数据,每个存储设备都被指定全局唯一标识,数据在物理存储设备6上的分布是由下述集群卷管理单元负责的。上述光纤/以太网交换机5为网络连接设备,用来转发来自服务器集群2通过FC(Fiber Channel)或TCP/IP协议包裹的小型计算机系统接口SCSI数据访问命令,上述服务器集群2和SAN网络共享存储目标器3通过光纤总线适配器或以太网网卡与光纤/以太网交换机5相连。

[0066] 提供容灾服务的容灾中心端的各节点提供容灾服务,如图2所示,本实施例中容灾中心端包括:

[0067] 容灾服务守护进程01,负责监听端口,接受来自客户端的容灾请求,对不同的请求作不同处理;在备份和恢复过程中,解析备份请求或恢复请求,转换为备份数据管理单元02识别的写入或读出命令,生成、维护和查询索引文件中内存映像和文件系统数据映像间关系;

[0068] 备份数据管理单元02,对于用户的每一次备份请求,备份数据管理单元会向下层机制发起生成相应的内存映像卷和数据映像卷的请求,并负责在用户空间管理这些不同的逻辑卷,为不同客户端不同时间点备份的查询在内存中建立高速索引,例如为代表各个卷的数据结构建立hash索引,还要记录系统相关数据,如虚拟卷的组成信息、属性和ID等,提供对各卷的定位查找、新建、删除、扩容等操作的接口;

[0069] 索引文件管理单元03,备份过程中,其负责生成与维护内存映像卷和数据映像卷的索引XML文件,并将其存放在容灾中心端的文件系统中,恢复时通过服务器启动过程中读取和分析索引XML文件,在内存中建立历史备份数据库,该数据库用来存放请求时间和卷ID的映射关系,生成高速索引,便于快速响应应用的数据请求,因此该单元只是建立请求时间<->卷ID映射的关系。

[0070] 备份数据管理单元02与索引文件管理单元03的关系为:在恢复过程中,索引文件管理单元03先执行,备份数据管理单元02后执行,索引文件管理单元03由时间来定位逻辑卷的ID,完成用户层面到系统层面的映射,备份数据管理单元02再由ID来查询其代表的

卷的数据结构,做的是纯系统层面的事,找出卷的属性信息。

[0071] 集群卷管理单元 04,虚拟出共享存储池供所有节点使用,负责备份数据管理单元 02 生成相应的内存映像卷和数据映像卷的存放和管理,具体为完成对海量网络共享存储池元数据的加载,通过分析元数据,建立完全备份虚拟卷、增量备份虚拟卷之间的依赖关系,完成对各内存映像备份虚拟卷、完全备份虚拟卷、增量备份虚拟卷快速索引结构的舒适化;同时,向上层提供符合 Posix 语义的统一读写接口,向内核驱动传送卷的属性更改信息,同时,该模块还负责同集群卷守护进程 07 通信,告知卷的更改;

[0072] 集群卷驱动单元 05,负责维护备份数据管理单元 02 生成相应的内存映像卷和数据映像卷的逻辑数据块与物理数据块映射信息,各逻辑卷读写命令的调度和派发,集群卷驱动单元 05 下层是物理设备驱动单元 06;

[0073] 物理设备驱动单元 06,通过存储区域网络 SAN 与存储池连接。

[0074] 集群卷守护进程 07,与集群管管理单元 04 通信,负责维护服务器集群的状态,保证集群中的有效节点数,同步对存储池中共享卷所作的修改。

[0075] 本实施例中容灾服务守护进程 01 与备份数据管理 02 配合对备份请求和恢复请求所作的处理分别为:

[0076] 对于备份请求,首先按照其请求类型,即全备份、增量备份、内存映像备份做不同处理:对于全备份,需要生成完全备份虚拟卷,即以逻辑卷的形式存放某一时间点时客户端系统的完全数据映像,以及对该虚拟卷的按序写入;对于增量备份,需要生成增量备份虚拟卷,仅仅记录自最近一次备份以来的少量数据变化,同时写入增量数据并建立数据的位置映射关系,即其在增量卷中的位置与其在被备份卷中的位置间的对应;对于内存映像备份,需要生成内存映像备份虚拟卷,以逻辑卷的形式存放某一时间点时系统的内存映像。

[0077] 对于恢复请求,要根据恢复请求中请求恢复数据的类型、时间点、位置等信息定位存储池中的具体虚拟卷,然后通过集群卷管理单元 04 提供符合 Posix 语义的统一读写接口对该卷进行访问,读取或写入指定数据块。

[0078] 客户端 1 是任意的待保护计算机,为了实现容灾,客户机必须主动向容灾中心端提出容灾申请,并告知相关参数,例如容量大小、备份类型等;客户端 1 的原有系统无需做任何修改,在客户端 1 上创建一个宿主系统和相应的软件环境,并将待保护的客户系统包容在一个虚拟环境中,客户端中各单元的结构如图 3 所示,客户端 1 主要包括:

[0079] 虚拟机环境构建单元 21,通过在客户端宿主系统上建立独立的进程组与会话,封装用户需要保护的应用及相关资源,构建出虚拟机环境,提供虚拟的终端和界面,保证用户正常使用,同时,虚拟机环境支持对其内部进程的内存映像进行冻结和迁移,这是实现应用无容灾的基础;

[0080] 内核拦截单元 22,用于监测虚拟机环境的每次读写请求,查询和维护记录了数据恢复情况的恢复数据位图,该内核拦截单元 22 被加载在内核的通用块设备驱动层,恢复数据位图存放在内核空间;

[0081] 虚拟机冻结单元 23,用于在内核拦截单元 22 监测到虚拟机环境中的进程有读写请求时,查询恢复数据位图,若要读写的数据未恢复,则在内核态发起虚拟机环境冻结操作;

[0082] 虚拟机解冻单元 24,用于在内核拦截单元 22 监测到虚拟机环境中的进程有读写

请求要读写的数据写入本地存储介质后,发起解冻虚拟机的操作,使虚拟机环境中的各个进程恢复运作;

[0083] 请求转发守护进程 25,用于与容灾中心端通信,在虚拟机环境下发出客户端备份或恢复请求并传送、接收数据,集中完成数据请求管理、调度实现容灾恢复,本实施例中请求转发守护进程通过开启一个本地侦听端口,接受本地的备份或恢复请求,当请求到来,首先建立连接;然后根据请求类型将其挂入备份任务或恢复任务的队列;同时,监听端口还负责监听来自内核拦截单元 22 传过来的数据块恢复请求。

[0084] 本实施例中请求转发守护进程 25 包括备份子进程 251 和恢复子进程 252,其中:

[0085] 备份子进程 251,用于接收用户发来的备份请求,并生成备份任务,通过遍历备份任务队列,根据每个备份任务的需求与容灾中心端建立一个连接,并转发该任务的后续请求到服务器端,同时根据各个任务的优先级集中对数据备份请求进行调度和流量控制;

[0086] 恢复子进程 252,用于接收用户发来的恢复请求,并生成备份任务,通过遍历恢复任务队列,根据每个恢复任务的需求与容灾中心端建立连接,接受本地连接发起方以及内核拦截单元发来的数据块恢复请求,将其维护在数据块恢复请求队列中,记录服务器端传送回来的数据,写入本地存储介质,同时对来自内核拦截单元的数据请求,恢复子进程还负责其回调函数的执行,完成数据恢复,并根据各个任务的优先级集中对数据块恢复请求进行调度和流量控制

[0087] 实现中,请求转发守护进程 25 采用了大循环的程序架构,如图 4 所示,当无事件发生时,整个进程处于阻塞侦听状态,所有的接口形成一个资源池,包括等待连接的端口、备份任务的数据输入端口、恢复任务的恢复数据接收端口等等,一旦有数据写入或请求等事件发生,则进程依序扫描各个端口,处理发生的事件;备份应用和恢复应用由用户发起,可以有多个,每个应用对应一次备份或一次数据恢复,对同一个虚拟机,同一时刻只能存在一个任务;备份任务顺序提交要备份的数据块给请求转发守护进程;恢复任务按照一定的恢复策略提交数据块恢复请求,同时需要查询恢复数据位图,避免重复提交。

[0088] 本实施例中基于虚拟机的按需增量恢复容灾方法流程图,该方法主要包括步骤:

[0089] s100,客户端 1 和容灾中心端初始化,该初始化过程具体为加载客户端 1 和容灾中心端中的上述各单元模块,在客户端 1 上构建虚拟机环境;

[0090] s101,客户端 1 通过备份子进程 251 接收用户发起的备份请求,生成备份任务,与容灾中心端建立一个连接,并转发该备份任务的后续请求到容灾中心端,该步骤中备份子进程 251 生成备份任务的方法为:启动多个备份应用,这里的备份应用由用户发起,可以有多个,每个备份应用对应一次备份请求,由设置参数指明备份的类型、待备份的文件系统或卷对象、备份数据块的大小等,向请求转发守护进程 25 的监听端口发出备份请求即可由备份子进程 251 创建一个备份任务;

[0091] s102,容灾中心端由容灾服务守护进程 01 通过监听端口接受客户端 1 的转发过来的备份请求,并将备份请求解析后通知备份数据管理 02;

[0092] s103,备份数据管理单元 02 根据备份请求生成客户端 1 虚拟机环境下相应的内存映像卷和数据映像卷,由集群卷管理单元 04 以逻辑卷形式存储到共享存储池,完成备份过程;

[0093] s104,客户端 1 启动恢复流程,主要包括:

[0094] (1) 设置恢复参数,向请求转发守护进程 25 的监听端口发出恢复请求 ;(2) 通知内核拦截单元 22 初始化恢复数据位图,并将位图通过内存映射开放出来,供请求转发守护进程 25 的恢复子进程 252 读取和更改,客户端的恢复子进程 252 接收用户发起的恢复请求,生成恢复任务,与容灾中心端建立一个连接,并转发该恢复任务的后续请求到容灾中心端,该步骤中恢复子进程 251 生成恢复任务的方法为 :根据用户的需求,启动恢复应用,这里的恢复应用由用户发起,可以有多个,每个恢复应用对应一次恢复请求,设置参数指明要恢复的时间点、待恢复的内存映像和文件系统映像、数据块的大小等,向请求转发守护进程的监听端口发出恢复请求即可由恢复子进程创建一个恢复任务 ;

[0095] s105,容灾中心端由容灾服务守护进程 01 通过监听端口接受发送过来恢复请求 ;

[0096] s106,索引文件管理单元 03 根据恢复请求检索和定位客户端虚拟机环境在指定时间点的内存映像卷,并由容灾服务守护进程 01 发送到客户端 1 ;

[0097] s107,客户端 1 的恢复子进程 252 接收内存映像卷,获取虚拟机环境在指定时间点的内存映像,其中包含进程树结构、各进程虚存空间的映像、文件系统挂载点等等 ;

[0098] s108,客户端 1 根据获取的内存映象进行内存映像重构,恢复客户端指定时间点虚拟机环境的正常运行状态,并行执行步骤 s109 和 s109' ;

[0099] s109,在客户端开启内核通用块设备的读写请求拦截功能,即启动内核拦截单元 22 的功能,当虚拟机环境中的进程访问文件系统或数据库时首先查询恢复数据位图,这里的文件系统只是客户端的应用,也可以是数据库,如果当前请求数据已经通过恢复子进程从容灾中心端读取过来,则直接访问 ;否则执行步骤 s110 ;

[0100] s110,客户端直接在内核态发起虚拟机环境的冻结操作,使隶属于虚拟机的各个进程陷入冻结,不再接受调度,然后向请求转发守护进程的监听端口发出对该请求数据的恢复请求 ;

[0101] s111,请求转发守护进程在收到来自内核拦截单元的数据块恢复请求后,将该请求挂载到步骤 s104 所建立恢复任务的数据块恢复请求队列中,并赋予其最高优先级和用来解冻虚拟机的回调函数,同时,根据相应的预取策略,恢复子进程也会找出与缺失数据较相关的数据块,生成相应请求并赋予高于其它恢复请求的较高优先级,然后按照优先级发送这些请求,并等待服务器端的响应 ;

[0102] s112,当恢复子进程接收到来自服务器端的待恢复数据块后,首先更新恢复数据位图,标志该数据为“已恢复”,并将该数据写入本地存储介质 ;

[0103] s113,发起解冻虚拟机的操作,使虚拟机环境中的各个进程恢复运作 ;

[0104] 在步骤 s113 之后,通过查询位图判断是否所有数据已恢复完毕,若是,则关闭内核拦截模块,释放位图,恢复正常读写,若没有,则重新返回到步骤 s109 进行恢复工作。

[0105] s109',由步骤 s104 所建立的恢复任务会按照既定的策略,缺省为顺序,在后台不断向恢复子进程发送数据块恢复请求,同时需要兼顾的原则是 :(1) 如果通过查询恢复数据位图,该数据块已经被标志为“已恢复”,则直接跳过此块,尝试发送策略指定的下一待恢复数据块 ;(2) 后台恢复请求尽可能不干扰内核拦截模块发出的数据块恢复请求,即通过设置优先级手段实现。本实施例中该步骤具体包括以下步骤 :

[0106] s109a,选择后台恢复策略 ;

[0107] s109b,根据恢复策略确定下一要恢复的数据块 x ;

[0108] s109c,通过查询位图判断该数据块 x 是否已恢复,若是,执行步骤 s109f,若否,执行步骤 s109d;

[0109] s109d,向恢复子进程发送请求获取数据块 x 的恢复请求,恢复子进程将该请求挂入恢复队列,并与容灾终端建立连接,发送恢复请求到容灾中心端,容灾中心端通过恢复请求查找并发送备份的数据块 x 到客户端;

[0110] s109e,客户端的恢复子进程收到数据块 x,然后更新位图,并将数据块 x 写入本地存储介质;

[0111] s109f,通过查询位图判断是否所有数据已恢复完毕,若是,结束,若否,返回执行步骤 s109b。

[0112] 该方法中由步骤 s109 开始的恢复流程和后台恢复流程即步骤 s109',都是在进行内存映象重构后进行数据块恢复的恢复任务,只是优先级不同,由步骤 s109 执行的恢复流程是在恢复过程中对要访问的数据的恢复,为了满足用户的及时访问,因此优选恢复,而在没有来自内核拦截模块的数据块恢复请求时,是按恢复策略进行后台数据块恢复的。

[0113] 本实施例中步骤 s108 中进行内存映像重构,恢复客户端指定时间点虚拟机环境的正常运行状态具体包括步骤:

[0114] s108a,派生一个内核进程 root_task,该 root_task 表示内核进程为始祖,修改内核进程 root_task 内核栈,令内核进程 root_task 在系统调用退出时切换为一个缺省的自恢复进程,该进程派生出其子进程,然后再次修改内核栈,从内存映像中拷贝客户端操作系统根进程的用户态堆栈、虚存映像等,构造第二次系统调用退出,进而切换为指定时间点时的用户态根进程;

[0115] s108b,由上述自恢复进程派生出的子进程依然为自恢复进程,这些进程根据内存映像中的进程树结构按从根到叶的顺序依次派生出新的子进程,直到内存映像中的所有进程都被派生出来为止,设置各个进程为深度睡眠状态;

[0116] s108c,在客户端另启动一个独立进程,该进程位于虚拟机环境外,该独立进程根据用户的指令将虚拟机环境中所有进程的状态按照内存映像中的记录进行重置,从而恢复虚拟机环境的正常运行状态。

[0117] 本实施例中在客户端 1 上构建虚拟机环境的方法具体包括步骤:

[0118] s100a,扩充客户端 1 操作系统内核中进程的结构,使之增加名称为 vm_context_info 的域,域 vm_context_info 代表进程所处的虚拟机进程上下文环境,其中包括虚拟机的 ID,该环境中的进程列表,根文件系统对应得设备和挂载点,以及数据统计信息;

[0119] s100b,虚拟机环境初始化,开启一个用户态进程 root_thread,完成终端输入输出设备的切换,进而通过系统调用切入内核态,主要完成如下工作:初始化虚拟机中进程组的调度机制;初始化虚拟机专用伪文件系统,即将本地物理文件系统的一个子树进行二次挂载,并虚拟出新的超级节点;初始化虚拟机相关系统调用等;

[0120] s100c,将上述进程切换到用户空间,在虚拟机上下文环境中执行客户端操作系统标准根进程的可执行代码 /sbin/init,使上述用户态进程 root_thread 成为该虚拟机环境的根进程,然后派生出相关子进程,创建相关系统服务;

[0121] s100d,建立本地操作系统中的进程上下文与虚拟机进程上下文的切换机制,并在虚拟机环境下执行需要作容灾保护的应用程序,即执行上述步骤 s101 ~ s113。

[0122] 本实施例中虚拟机进行冻结的方法包括以下步骤：

[0123] s113a,根据要保护的应用,切换到其所属的虚拟机进程上下文环境,进而遍历该环境中的进程列表,置位冻结位,同时考虑几种特殊情况,例如刚刚 fork 完成而没有执行 exec 的进程、被 trace 而处于停止状态的进程,等待一定时间重新尝试冻结,对于停止的和僵尸,则直接忽略;

[0124] s113b,唤醒隶属于虚拟机的各个进程,给其发送一个伪信号,让其陷入冻结处理,不再接受调度,等待各个进程都处于冻结状态;

[0125] s113c,使用最新的现有同步机制Read Copy Update来设置内存屏障,通过同步操作等待各底层 I/O 驱动或网络驱动数据操作的结束。

[0126] 当用户需要同时对多个系统和应用提供容灾保护时,在一台客户端 1 节点上可能存在多个备份任务和恢复任务,本实施例中备份子进程 251 和恢复子进程 252 对不同重要程度的任务赋予不同的优先级,并按照优先级进行调度。调度的基本原则是:优先级越高占用带宽越高,通过让传送的每一帧包含较多的高优先级数据块请求来达到按优先级调度的目的,具体实现中,来自内核拦截单元的缺失数据块的请求被赋予最高优先级,根据相关策略预取的与缺失数据块相关的数据块请求被赋予较高优先级,而恢复任务发来的请求被赋予低优先级,其获取和恢复流程在后台运行。

[0127] 客户端 1 中内核拦截单元 22 在向请求转发守护进程 25 发出缺失数据块的恢复请求后,还会进行预取,即提交与缺失数据块物理位置邻近的各个块的恢复请求,邻近位置由设定的地址偏移阈值 offset 决定,这是一个需要动态调优的参数,预取数据命中会增加其绝对值,反之则减小。

[0128] 恢复任务按照后台数据恢复策略执行恢复流程,其恢复策略用来决定数据恢复的顺序,其原则是:尽量先获取应用读写频繁且运行时经常使用的数据块,通过先传送这些敏感数据,使得虚拟机环境中的应用启动后,大部分数据都能预先准备好,使内核拦截单元 22 发现缺失数据块的几率尽可能的小,减少冻结虚拟机环境、获取数据、解冻等流程带来的开销。本实施例的实现方法是:对每一数据块设置了敏感度,数据传输的顺序由敏感度决定;敏感度通过应用正常运行时,统计每个数据块在一个固定大小时间窗口 w 内的读写次数而获得,敏感度的数值与具体应用以及具体时刻有关,其计算公式如下:

$$S_w = Count_w * (1 - \alpha) + S_{w-1} * \alpha$$

[0130] 其中 S_w 代表当前的敏感度, S_{w-1} 代表前一个时间窗口的敏感度, $Count_w$ 为当前窗口的读写计数, α 称为历史因子,用来反映历史敏感度与当前敏感度之间的关联;敏感度相关数据结构为敏感度数组和读写计数位图,它们都被保存在应用所处的虚拟机环境的中,每隔时间 w 需要重新计算敏感度,更新数组和清空位图,开始新一轮计数;当备份时,敏感度数组作为内存映像的一部分也会被备份到上述容灾中心端。

[0131] 本实施例中该系统实现中还对如下问题作了相应处理:(1) 客户端 1 错误处理:当备份和恢复过程中连接应用、请求转发守护进程以及服务中心的容灾服务守护进程的插接口 Socket 出现校验错误和意外中断时,采用了错误重传和超时释放插接口的方法;(2) 位图的压缩:对于大容量的数据映像,其位图消耗的内存资源也不少,为了减少内存消耗,需要按照位图的特点做压缩,本实施例采用了类似游程码的压缩方式,这是因为大多数连续数据块都是连续更新的,位图中连续的‘1’和连续的‘0’比较多;(3) 位图的并发访问:

由于位图可能被内核拦截模块和请求转发守护进程并发访问,为了保证一致性同时提高并行度,使用了 Linux 2.6 内核中的读写锁 RCU(rwlock) 机制,在写操作更新数据的同时,暂不覆盖磁盘上原有的老数据,而是在内存中新开辟一块空间存放更新数据,这样达到新老数据的暂时并存;此时再发起的读写操作都定位到这个新开辟的空间,而之前还未完成的读操作也能够继续其对磁盘上的老数据的访问,该过程一直持续到所有对老数据的读访问全部结束;(4) 对于恢复请求,需要建立快速查询机制,当恢复数据到来时,需要定位恢复请求的相关数据结构,因此,对于每个恢复请求,本实施例除了将其通过指针链接成为一个双向链表,还在客户端内存中建立了请求 hash 表,根据请求号将各个请求放置到各个哈希桶 Bucket 中;(5) 保证恢复数据位图更新和磁盘数据更新的原子性:当恢复子进程接收到数据后,先写入位图,告知后续程序数据已经传送到,以便应用不会因为发现位图相应位为‘0’而冻结虚拟机带来额外开销,同时位图的设置也起到了 REDO 日志的作用,如果数据没能成功写入,接收到的数据会一直保留并被重写,直到数据顺利写入完成任务提交。

[0132] 本地的各种类型的客户端主机,首先,可以运行任意服务;其次,客户端主机会构建虚拟机进程上下文环境,通过该环境将要保护的应用打包,截取其一致的内存映像,同时并不影响应用的运行;最后,在客户端主机上集成虚拟机进程上下文环境,仅需购买第三方的安全容灾存储卡,不需要对其原有操作系统作任何改变。

[0133] 以上实施方式仅用于说明本发明,而并非对本发明的限制,有关技术领域的普通技术人员,在不脱离本发明的精神和范围的情况下,还可以做出各种变化和变型,因此所有等同的技术方案也属于本发明的范畴,本发明的专利保护范围应由权利要求限定。

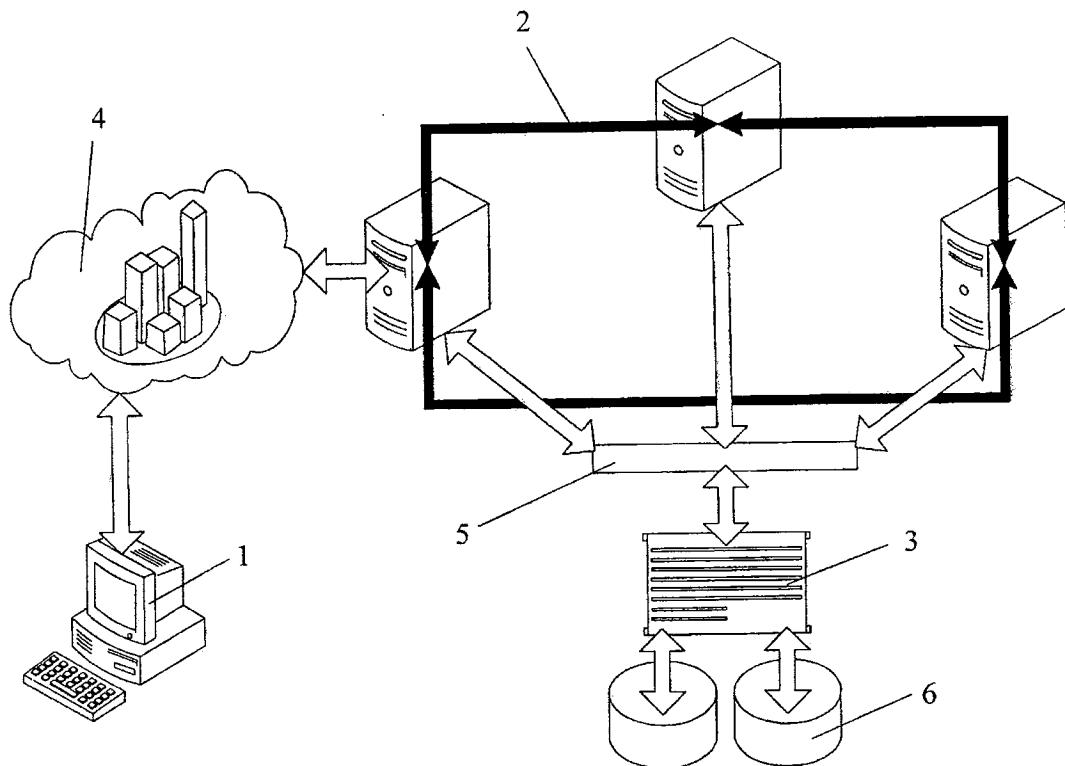


图 1

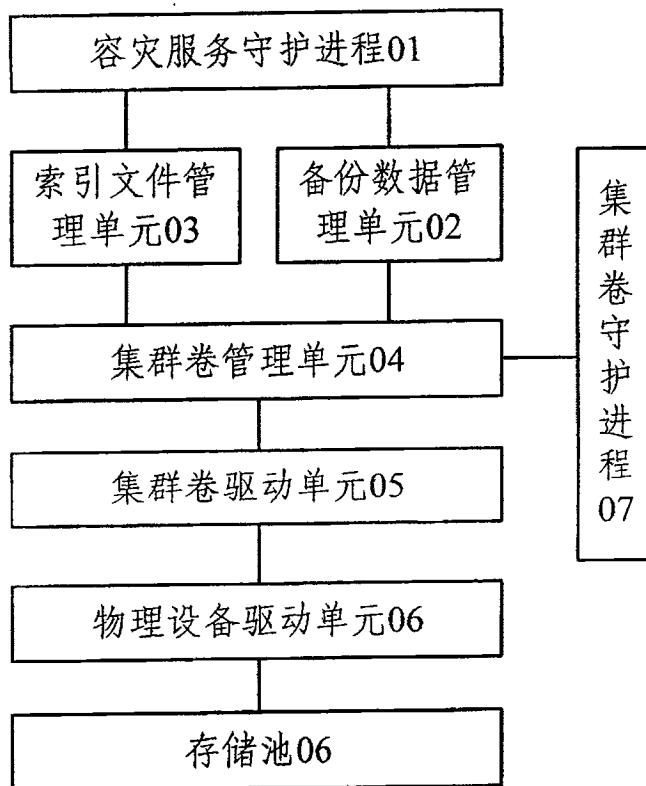


图 2

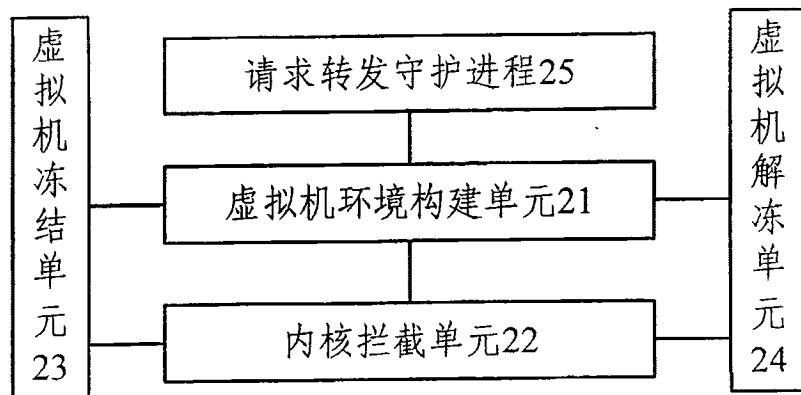


图 3

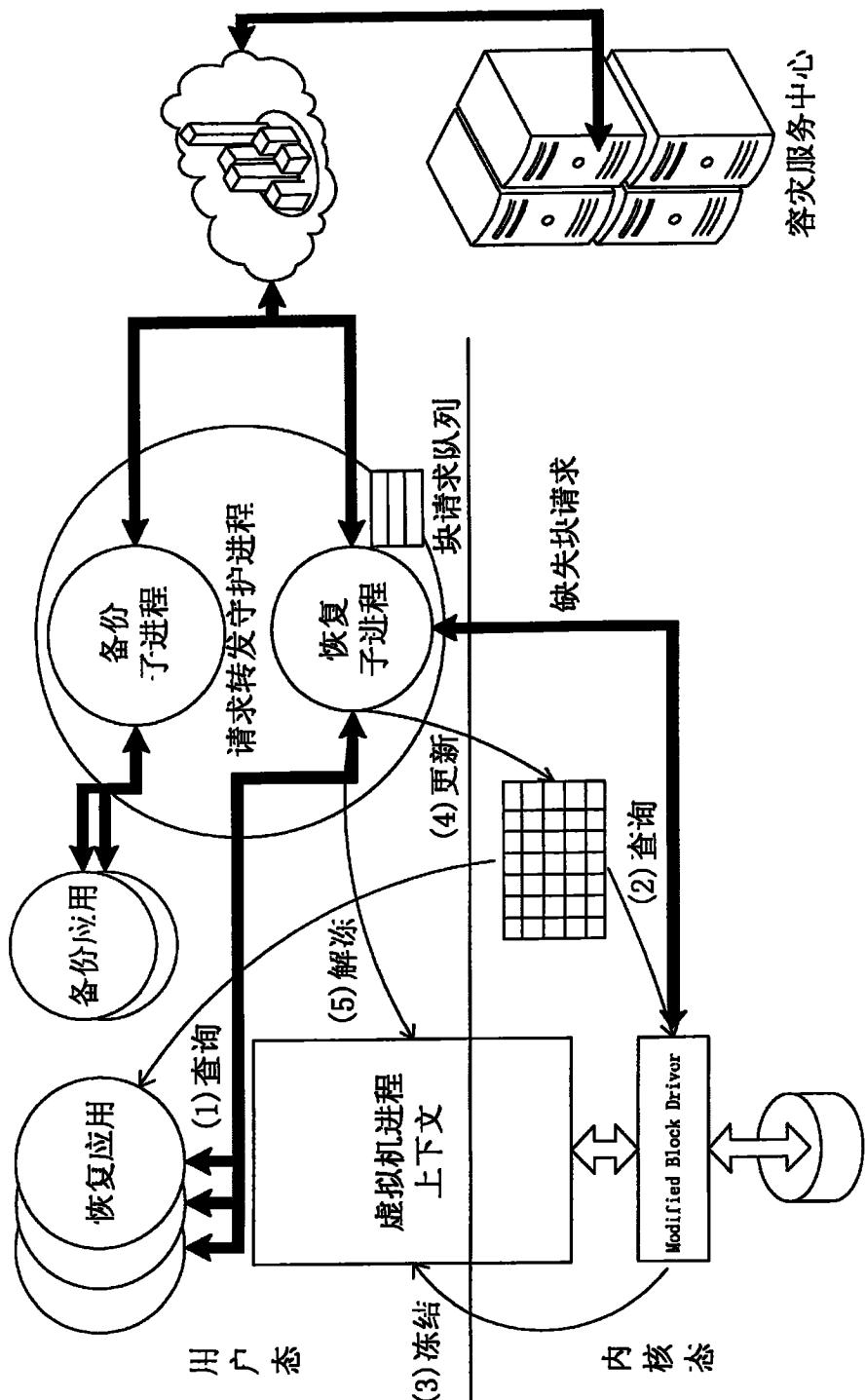


图 4

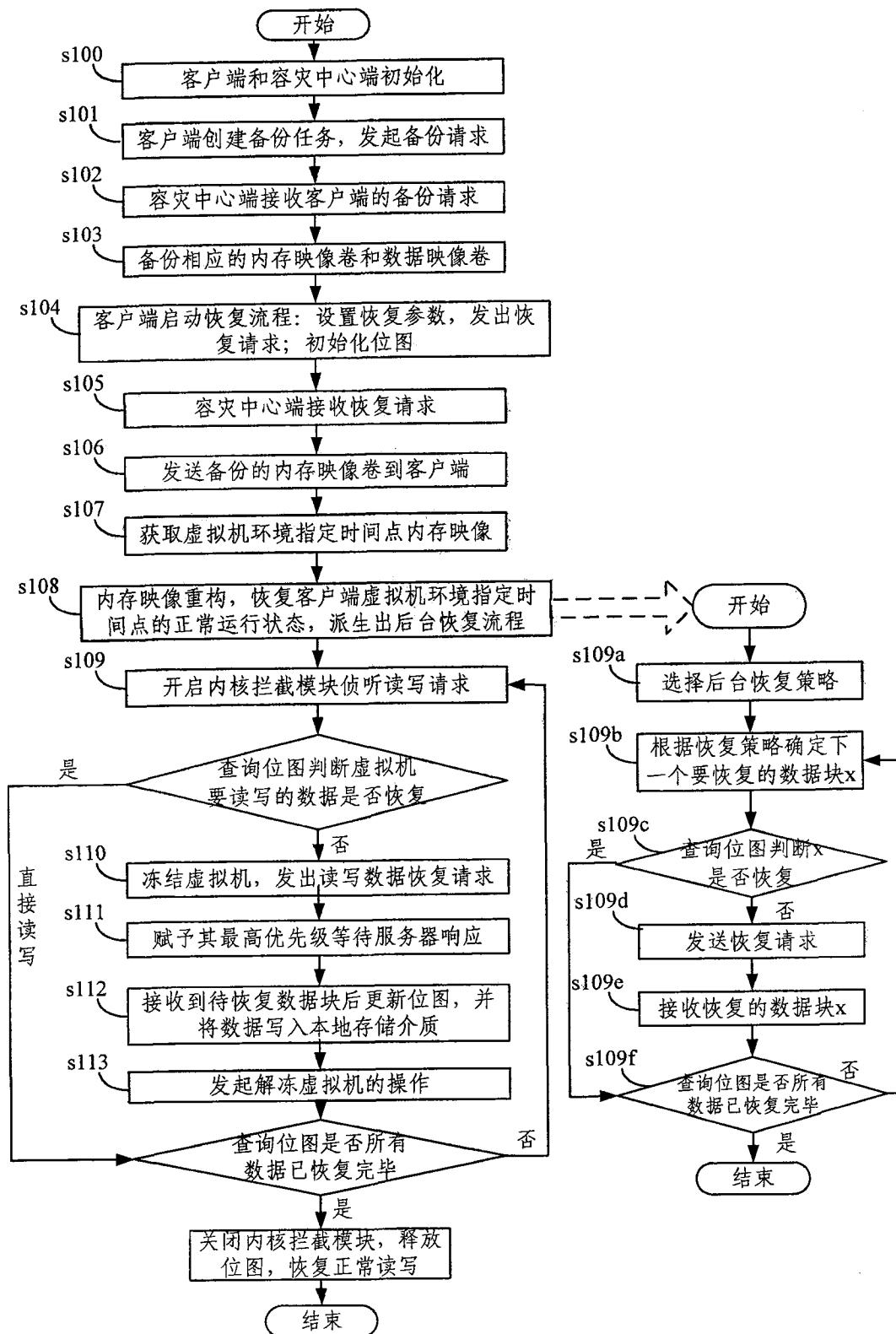


图 5