(19)国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 115997190 A
(43)申请公布日 2023.04.21

(21)申请号 202180053403.0
(22)申请日 2021.06.16
(30)优先权数据
16/918,822 2020.07.01 US
(85)PCT国际申请进入国家阶段日
2023.02.28
(86)PCT国际申请的申请数据
PCT/US2021/037532 2021.06.16
(87)PCT国际申请的公布数据
WO2022/005744 EN 2022.01.06
(71)申请人 甲骨文国际公司
地址 美国加利福尼亚
(72)发明人 D·A·小格雷夫斯
A·D·布鲁内尔 F·S·格鲁夫
(74)专利代理机构 中国贸促会专利商标事务所有限公司 11038
专利代理师 刘玉洁
(51)Int.Cl.
G06F 3/06(2006.01)

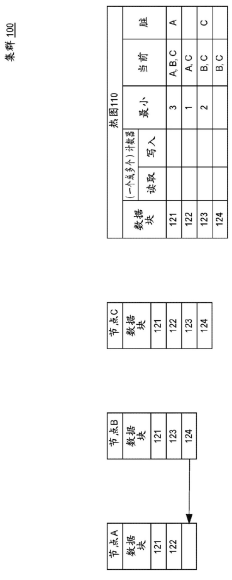
权利要求书2页 说明书18页 附图7页

(54)发明名称

完全一致的高效非本地存储集群文件系统

(57)摘要

热图用于识别数据块系列在具体时间的访问模式,以优化复制节点集合的大小,用于降低访问延迟的目的。在实施例中,热图用于诸如当复制节点崩溃并被替换时,强制执行每个数据块的最小复制。在实施例中,热图用于诸如在需求峰期间和之后调整数据块的最小复制。在实施例中,每个数据块在集群的最小数量的相应复制节点上被复制。第一节点请求访问数据块。基于请求访问,热图被修改,并且数据块被复制到第一节点。基于热图,针对至少一个数据块的相应复制节点中的最小节点数量被调整。



CN 115997190 A

1.一种方法,包括:

将多个数据块中的每个数据块复制到多个节点中的最小数量的相应的一个或多个复制节点上;

由所述多个节点中的第一节点请求访问所述多个数据块中的一个或多个数据块;

基于所述请求所述访问:

修改热图,以及

将所述一个或多个数据块复制到第一节点;

基于所述热图调整针对所述一个或多个数据块中的至少一个数据块的相应的一个或多个复制节点的所述最小数量。

2.如权利要求1所述的方法,还包括,基于所述请求所述访问,将第一节点添加到针对所述一个或多个数据块的所述相应的一个或多个复制节点。

3.如权利要求1所述的方法,还包括,响应于所述请求所述访问,基于所述访问的类型,针对所述一个或多个数据块授予:

防止所述多个节点修改所述一个或多个数据块的读取锁,或

提供第一节点的独占访问的写入锁。

4.如权利要求3所述的方法,还包括释放所述写入锁,释放所述写入锁包括:

在针对所述一个或多个数据块的所述相应的一个或多个复制节点中仅保留第一节点,以及

将所述一个或多个数据块复制到所述多个节点中包括第一节点的所述最小数量的相应的一个或多个复制节点。

5.如权利要求3所述的方法,其中:

所述针对所述一个或多个数据块中的至少一个数据块授予所述读取锁不发生在所述授予所述写入锁和第二次向第二节点授予针对所述至少一个数据块的所述写入锁之间;

所述方法还包括响应于第二节点释放所述写入锁,选择用于复制所述至少一个数据块的所述多个节点中包括第二节点且不包括第一节点的所述最小数量的相应的一个或多个复制节点。

6.如权利要求3所述的方法,其中所述调整所述热图是基于所述访问的所述类型的。

7.如权利要求1所述的方法,还包括响应于所述请求所述访问,记录以下的标识符:第一节点,以及所述一个或多个数据块中的至少一个数据块。

8.如权利要求1所述的方法,其中所述调整所述热图包括哈希。

9.如权利要求1所述的方法,其中所述调整针对所述至少一个数据块的相应的一个或多个复制节点的所述最小数量包括针对所述一个或多个数据块中的所述至少一个数据块:

异步,

通过贝叶斯网络预测,

基于写入访问请求的量而减少相应的一个或多个复制节点的所述最小数量,和/或

基于读取访问请求的量而增加相应的一个或多个复制节点的所述最小数量。

10.如权利要求9所述的方法,其中所述通过所述贝叶斯网络预测包括计算以下的概率:

两个数据块将被一起访问,

一个数据块将被访问,因为另一个数据块已被访问,

第一节点将访问特定数据块,和/或

锁将在持续时间内释放。

11.一个或多个非暂时性计算机可读存储介质,其存储指令,所述指令当由一个或多个处理器执行时,使得如权利要求1-10中任一项所述的步骤被执行。

完全一致的高效非本地存储集群文件系统

[0001] 相关案例

[0002] Ranjit Noronha等人在2007年并行处理国际会议(ICPP 2007)上的相关非专利文献(NPL)“Designing NFS With RDMA for Security,Performance and Scalability”整体并入本文。

技术领域

[0003] 本发明涉及使用访问活动的热图来管理跨多个存储节点的数据复制。

背景技术

[0004] 对于文件集群,存在管理共享存储和集群成员资格的各种方式。一些方法使用共享存储和全连接的互连来提供全连接的集群。其他集群类型涉及本地存储和松散连接的集群。这些允许集群的扩展超出全连接的集群通常所提供的程度,但可能具有数据表示的不同语义。

[0005] 在文件系统级别,分布式文件系统可以通过至少两种方式实现:

[0006] • 无一致性——文件被完全复制到每台机器上,从而防止集群的总存储大于单台机器的存储。没有机器知道另一台机器对特定文件的更新。如果两台机器同时修改文件,则最后的修改获胜。

[0007] • 有限的一致性——文件被复制到机器的子集,从而允许集群的存储大于任何一台特定机器的存储。机器知道其他机器的修改,但不提供分布式锁定或其他机制来防止两台机器同时修改相同的文件,因此,最后的修改获胜。

[0008] 一般来说,分布式文件系统有两个不幸的特性:

[0009] • 对非本地文件访问慢,因为文件必须传送到访问它们的节点,或者客户端必须被重定向到托管文件的节点。

[0010] • 由于集群服务器的松散的集群一致性,无法跨集群跟踪访问模式。

[0011] 当前的解决方案具有各种相应的缺点,诸如:

[0012] • 缺乏或只有有限的便携式操作系统接口(POSIX)合规性;

[0013] • 两步复制——只有在数据被写入盘后,它才被复制。大多数复制实现使用远程同步(rsync)。Rsync有三个主要缺点:a)仅支持整个文件,b)需要额外的网络往返来检测文件的哪些部分已改变,以及c)文件的时间戳具有一秒的时钟分辨率,这意味着两台机器不得在相同的时钟秒内改变相同文件的不同部分,因为假定相同的时间戳指示相同的改变;

[0014] • 低效的复制——无论那些数据块被改变,整个文件都被复制;

[0015] • 无法对本地节点上的访问作出反应——没有针对访问模式的调整;

[0016] • 在一些情况下,诸如NFS之类的公共协议不可用——客户端必须使用特殊的翻译器库以使用(speak)分布式文件系统的语言;以及

[0017] • 仅在某些场景下有用——不是通用的。为了让其在特定场景中有用,数据和访问模式必须正合适,或者益处必须超过缺点。

[0018] 本节中描述的方法是可以追寻的方法,但不一定是先前已经设想或追寻的方法。因此,除非另有说明,否则不应假定本节中描述的任何方法仅仅因为它们包含在本节中就成为现有技术。

附图说明

[0019] 在附图中:

[0020] 图1是描绘示例集群的框图,该示例集群通过通信网络或互联网络将数据块复制到节点的相应子集。

[0021] 图2是描绘将数据块从一个节点复制到另一个节点的示例实现方式的流程图。

[0022] 图3描绘了随着各种场景的进行和交互可能发生以创建和丢弃数据块的副本的示例步骤序列。

[0023] 图4描绘了读取器节点和数据块的示例读取生命周期。

[0024] 图5描绘了其中复制和热图可以彼此影响的示例计算机过程。

[0025] 图6是图示本发明的实施例可以在其上实现的计算机系统的框图;

[0026] 图7是图示可用于控制计算系统的操作的基本软件系统的框图。

具体实施方式

[0027] 在以下描述中,出于解释的目的,阐述了许多具体细节以提供对本发明的透彻理解。然而,将明显的是,可以在没有这些具体细节的情况下实践本发明。在其他情况下,众所周知的结构和设备以框图形式示出以避免不必要地模糊本发明。

[0028] 总体概述

[0029] 将本文的全连接集群与本地存储相结合提供了有趣的机会,诸如其存储可以增长到大于集群中任何一台机器的存储但不需要在节点之间直接共享存储的集群。本文中的集群,当针对对等数据转移是完全互连的时,保持适当的文件更新修改并防止将破坏或丢失数据的对文件的冲突修改。本文中的集群保证了节点之间的文件系统一致性而没有完全复制的开销。

[0030] 本文中的集群文件系统提出了一种逻辑文件系统,客户端可访问该逻辑文件系统而无需了解其文件存储在哪里。本文中的技术让集群服务器的存储组合成更大的逻辑文件系统,客户端通过诸如网络文件系统(NFS)、本地访问、服务器消息块(SMB)或其他方法之类的方便的手段访问该更大的逻辑文件系统。

[0031] 为集群文件系统提供技术,该集群文件系统聚合集群中节点的本地存储以表示符合Posix的存储解决方案,该解决方案具有对诸如文件、文件的部分和/或单个数据块或数据块系列之类的持久数据对象的高效复制。复制提供了重要的好处,诸如数据局部性以减少网络流量和冗余以避免数据丢失。冗余可以避免否则可能妨碍复杂系统的单点故障。例如,即使当主动参与的节点经历盘崩溃、软件故障或诸如磁盘空间、暂时存储器或处理器或输入/输出(I/O)带宽之类的资源耗尽时,本文中的技术也提供高可用性(HA)。复制可以向相同或不同数据中心中的诸如用于数据仓库、数据网格或事务集群(诸如弹性云)的高容量或复杂系统提供可靠性、可用性和可服务性(RAS)。

[0032] 在集群文件系统中,客户端与多个节点中的一些或全部交互。集群中的每个节点

维护本地存储，该本地存储在逻辑上与集群中的所有节点共享，即使没有物理共享。例如对于复制，本文中的技术不需要交叉安装的文件系统，诸如NFS。每个节点实现了可以持久化集群的数据块的相应子集的高速缓存。同样，每个数据块都被复制到集群的节点的相应子集。持久化相同数据块的副本的节点的子集在本文中被称为该数据块的复制节点。高速缓存一致性在复制节点之间维护。也就是说，数据块的每个复制节点维护数据块的一致副本。

[0033] 在实施例中，节点集群中的第一节点请求从存储数据块系列的副本的一组复制节点中的复制节点访问数据块系列。复制节点集合被配置为在它们相应的本地存储装置内存储数据块系列的副本。复制节点集合中的复制节点在接收到访问请求后，可以向第一节点提供数据块的副本。第一节点可以接收数据块系列的副本。一旦被接收，第一节点可以将数据块系列的副本持久化到第一节点上的本地高速缓存中。在实施例中，本地高速缓存包括易失性和非易失性存储装置，两者都存储如本文讨论的相同的数据块。

[0034] 节点集群中的每个节点利用集群管理软件被启用，集群管理软件被配置为管理数据块系列的锁和数据块系列的复制节点集合。在接收到数据块系列的副本后，第一节点上的集群管理软件可以在访问活动的热图中记录针对数据块系列的访问活动，该热图用于跟踪关于数据块系列的读取和写入访问以及分配的锁。集群管理软件然后可以确定是否调整复制节点集合中的被配置为存储数据块系列的副本的节点数量。响应于确定是否针对数据块系列来调整复制节点集合中的节点数量，集群管理软件可以调整复制节点集合中的节点数量。

[0035] 利用分布式锁管理器(DLM)层和全连接集群，访问模式的热图可被创建。如果通过DLM层请求访问文件或数据块的每个节点都被指派了唯一标识符(这可以是DLM的要求)，那么最频繁访问文件或数据块的节点的图可被动态更新。读取和修改操作可被单独跟踪，从而允许集群知道文件在哪里被访问最多以及以何种方式被访问。

[0036] 这些热图可以确保数据被复制到使用该数据最多的节点。各种启发式方法可被使用：对于N路复制，确定文件的特定数据块系列的前N个需求最多的节点——并确保所有这些顶部节点都包含数据副本。这有两个目的——通过复制创建数据冗余，并为对数据有最大需求的那些节点保持数据访问是本地的。其他算法和/或启发式方法可以附加地或代替地与热图一起使用以调优集群的性能。例如，诸如贝叶斯网络、预测系统和其他优化调优之类的人工智能(AI)算法可以确定各种文件或数据块的复制和存储设置。一个总体目标是实现低延迟写入和低延迟读取，这在其他方法中是对立的关注点。

[0037] 在实施例中，如果集群管理软件确定对数据块系列的读取访问请求的数量超过配置的阈值，则集群管理软件可以增加复制节点集合中的节点数量。例如，读取次数中的增加可能意味着在附加的复制节点上具有数据块系列的附加副本更高效，从而减少与将数据块系列从一个复制节点复制到不在复制节点集合中的节点相关联的延迟。

[0038] 然而，写入访问请求需要改变的数据块被复制到复制节点集合中的每个复制节点。如果复制节点集合大，则将新的写入复制到每个复制节点可能需要大量时间和资源，从而增加数据块系列的其他访问请求的延迟。如果集群管理软件确定对数据块系列的写入访问请求的数量超过了配置的阈值，则集群管理软件可以减少复制节点集合中的节点数量，以减少跨集群的写入数量以便减少延迟。

[0039] 热图可用于识别数据块系列在具体时间的访问模式，以优化复制节点集合的大

小,用于降低访问延迟的目的。在实施例中,诸如当复制节点崩溃并被替换时,热图用于强制执行每个数据块的最小复制。在实施例中,诸如在需求峰期间和之后,热图用于调整数据块的最小复制。

[0040] 在实施例中,每个数据块在集群的最小数量的相应复制节点上复制。第一节点请求访问数据块。基于请求访问,热图被修改,并且数据块被复制到第一节点。基于热图,针对至少一个数据块的相应复制节点中的节点的最小数量被调整。

[0041] 集群文件系统

[0042] 图1是描绘示例集群100的框图,集群100通过通信网络或互联网络将数据块121-124复制到节点A-C的相应子集,这些节点可以是相同或不同的网络元件类型,诸如机架服务器(诸如刀片)、大型机、存储设备、笔记本电脑、智能手机或虚拟机(VM)。数据块121-124各自具有固定量的数据存储容量。在实施例中,数据块121-124是相同或不同文件的连续或不连续的盘块。在实施例中,数据块121-124是数据库块和/或不是文件的一部分。

[0043] 数据块121-124可以单独地或以各种子集在节点A-C的相同或不同的相应子集上复制。例如,数据块121在所有节点A-C上复制,而数据块124仅在节点B-C上复制。不同的节点可能存储相同或不同数量的数据块。例如,节点C具有比节点A多的数据块。

[0044] 数据块121-124的相应系列或个体的最近访问频率被记录在热图110中,该热图110可以驻留在中央服务器或节点A-C中的一个、一些或所有的易失性或非易失性存储装置中。在所示实施例中,热图110具有单独计数(诸如在当前时间段期间)每个数据块的最近读取和写入的列。例如,当任何节点读取数据块122时,数据块122的读取计数器在热图110中递增1。

[0045] 在实施例中,热图没有用于单个数据块的(一个或多个)访问计数器。例如,可能有由多个数据块共享的(一个或多个)访问计数器,该多个数据块诸如:一系列数据块、具有冲突哈希码的数据块或一系列哈希码的数据块。例如,哈希函数可以基于诸如逻辑块地址(LBA)之类的数据块的标识符来计算数据块的哈希码。

[0046] 在实施例中,读取和/或写入计数器在相同或不同的相应时间段之间在热图110中被重置为零。在实施例中,读取和/或写入计数器根据相同或不同的线性或非线性冷却时间表或多或少地逐渐减少。

[0047] 在实施例中,读取和写入影响相同的计数器。热图110的实施例可以包含比所示的列更多或更少的列。在实施例中,一些所示的列附加地或替代地包含在除热图110之外的(一个或多个)软件组件中。

[0048] 如贯穿本文所讨论的,热图110包含可用于自动优化副本的数量和位置以保护数据和吞吐量的跟踪数据和指标。集群100中的哪个(哪些)组件请求或执行这样的自动副本优化可以取决于实施例。在本文讨论的大多数实施例中,执行自动副本优化的唯一或主要组件是当前访问数据块的节点,这需要检查热图110。

[0049] 在实施例中,附加地或代替地,锁管理器基于热图110执行自动副本优化。在实施例中,热图110是面向对象的,以提供数据和行为,诸如自动副本优化。使节点负责自动副本优化的优点是去集中化可以避免自动复制优化的单点故障,这使得集群100更加健壮。在这样的实施例或其中锁管理器提供自动副本优化的实施例中,热图110可以是被动数据结构。

[0050] 在各种实施例中,自动副本优化可以在各种时间发生。所有实施例都可以在重要

时间执行自动副本优化，诸如当：a)数据块被锁定、访问和/或解锁时，或b)当节点崩溃或以其他方式离开集群100时，如本文稍后讨论的。一些实施例是自治的并且根据如本文稍后讨论的各种时间表、触发和/或条件在附加的时间(诸如甚至当集群100以其他方式空闲时)执行自动副本优化。

[0051] 如上所述，在各种实施例中，主动节点、锁管理器或热图110可以基于热图110执行自动副本优化。集群100的自治地进行该动作的这样的组件在本文中被称为自治组件。集群100可以具有相同或不同种类的一个或多个自治组件。例如在实施例中，一些或所有节点和热图110可以是自治组件，其可以具有相同或不同的原因和逻辑以自治执行自动副本优化。

[0052] 在实施例中，在节点对数据块的访问期间，节点在热图110中记录多于递增的访问计数。热图110中还可记录进行访问的节点和数据块的标识符。在实施例中，热图110包含任何节点对任何数据块的最近访问的或多或少的详细日志。

[0053] 热图110的当前列指示包含相应数据块121-124的节点A-C的子集。例如，所有节点A-C都包含数据块121。如果节点已本地持久化数据块的副本，则该节点是数据块的复制节点。如图所示，所有节点A-C都是数据块121的复制节点，但节点A不是数据块123-124的复制节点。

[0054] 如本文稍后所讨论的，热图110的最小列在不同时间期间被使用。集群100确保每个数据块总是至少在最小数量的复制节点上被复制。在未示出的实施例中，所有数据块121-124具有相同的最小数量的副本。在所示的实施例中，不同的数据块具有不同的最小数量。例如，数据块122具有比所需更多的副本。

[0055] 各种实施例可以具有各种粒度(诸如一个数据块、一系列数据块和/或文件中的所有数据块)的数据访问锁。节点在访问(一个或多个)数据块时应该获取(一个或多个)锁。在实施例中，所有节点A-C都可以从锁管理器请求锁。各种实施例具有用于所有锁的锁管理器或用于每个锁的单独的锁管理器。在各种实施例中，锁定或解锁(一个或多个)数据块的请求被提交给由中央服务器或节点A-C中的一个、一些或所有托管的锁管理器。在实施例中，热图110和锁管理器是相同的组件。

[0056] 在访问数据块之前节点应该锁定数据块，无论数据块是否已经被本地存储在节点上。节点应始终访问数据块的其本地副本，即使这需要突然将数据块从另一个节点的易失性或非易失性存储装置复制到该节点的易失性或非易失性存储装置中。例如如图所示，节点A从节点B复制数据块124，即使节点C可以代替地提供相同的数据块124。

[0057] 为了决定节点B或C是否应该向节点A提供数据块124，各种实施例可以具有各种标准，诸如网络拓扑、工作负载和/或性能指标。在各种实施例中，节点A或锁管理器或热图110具有决定哪个复制节点应该提供数据块的逻辑。

[0058] 锁语义如下。如所示出的，对于不同的访问类型(诸如读取和写入)有单独的锁。在实施例中，诸如文件截断之类的(一个或多个)数据块的删除是写入的示例。写入锁提供对一个节点的独占访问。如锁列所示，“W:A”意味着节点A使数据块121写锁定。在实施例中，锁列是锁管理器的一部分而不是热图110的一部分。

[0059] 多个节点可以同时使相同数据块读取锁定。如所示出的，“R:A,C”意味着节点A和C两者使数据块122读取锁定。相同数据块不应该同时被写入锁定和读取锁定。如所示出的，数据块123-124被解锁。在实施例中，节点向其他节点和/或热图110内的记录通知该节点正

在请求读取锁或写入锁。

[0060] 获取写入锁的节点应该最终且明确地释放锁。读取锁可以如下被明确地或隐含地释放或破坏。即使对于已经被其他节点读取锁定的数据块，写入锁的获取也可被发起。当数据块的写入锁被获取时，诸如以下的多种反应可能出现。

[0061] 在实施例中，已经获取该数据块的读取锁的所有节点都被通知它们的读取锁现在被自动破坏。在实施例中，需要恢复读取数据块的节点应该重新获取读取锁，该读取锁直到写入锁被明确释放后才被授予，此时新版本的数据块可被读取。因此，在读取锁的重新获取前后观察到不同版本的数据块。在实施例中并且在一些情况下，读取器节点可以在没有锁的情况下，诸如在读取锁被破坏之后，决定继续使用来自陈旧本地副本的内容。例如，数据库会话的隔离级别可被配置用于可重复读取。

[0062] 在实施例中，已获取读取锁的节点可以升级到写入锁，这以与在没有已经具有读取锁的情况下获取写入锁相同的方式发生，并且这隐含地打破了读取锁。在实施例中，已经获取写入锁的节点可以降级为读取锁，这以与解锁写入锁并且然后获取读取锁相同的方式发生。

[0063] 授予锁的公平性防止需要数据块的节点饥饿。在实施例中，诸如通过使请求排队，以与针对相同数据块所请求的相同的时间顺序授予锁。释放锁的请求从不排队。在实施例中，降级锁在无需排队的情况下发生。

[0064] 最小复制

[0065] 本文中的技术合并两种复制。如本文稍后所讨论的，按需复制需要在需要但缺少数据块的节点上创建数据块的额外副本。如果相同数据块最终(无论是否同时)在集群的许多或所有节点上都被需要，则实施例最终可以将该数据块复制到集群的许多或所有节点。换句话说，许多或所有节点可以同时是相同数据块的复制节点，这确保了该数据块的充分复制。

[0066] 另一种复制是最小复制以在不管需求的情况下确保最小冗余。例如，当前可能所有节点都不需要数据块，但仍然应该将数据块持久化在少许节点上，以避免在任何节点将来需要该数据块的情况下数据丢失。由于需求复制，而不是最小复制，在许多节点上需求高的数据块通常将在集群中有充足的副本。

[0067] 当只有一个节点需要并写入数据块时，技术问题可能出现。本地写入数据块必然导致远程副本变得陈旧。由于其他节点对该数据块没有当前需求，因此需求复制不发生以传播修正后的数据块。缺乏复制的此问题通过最小复制来解决，为此其他问题出现，包括当在以下情况时决定哪些节点应该是复制节点：a)在低需求的情况下没有或只有一个节点需要数据块，或者b)在相反的高需求情况下，存在数据块的太多的陈旧副本，而无法用修正后的数据块替换所有陈旧副本。本文中的技术通过将最小复制集成到各种触发(诸如如下的写入锁的释放)中来解决用于最小复制的复制节点选择的这些问题。

[0068] 当数据块的写入锁被释放时，附加的活动如下发生，包括最小复制的强制执行。如上所解释的：a)如果节点缺少本地副本，则获取读取或写入锁的节点也接收并持久化副本，以及b)可以创建超过最小数量的数据块副本。当数据块的写入锁被释放时，包括获取写入锁的节点的最小数量的复制节点被选择为仍然是复制节点。在各种实施例中，热图110或获取和释放写入锁的节点选择剩余的复制节点。在实施例中，相同的同步调用路径(即，控制

流)包括:写入锁定、修改数据块、解锁、本地持久化、通知热图110和/或将修正复制到剩余的复制节点。

[0069] 选择剩余复制节点的标准可以包括或者可以不包括:诸如具有分层网络的网络拓扑、工作负载和/或诸如未使用的本地盘空间之类的性能指标。在实施例中并且如本文稍后讨论的,贝叶斯网络计算并比较各种概率以通过预测哪些复制节点可以最大化吞吐量和/或优化其他性能指标来选择剩余的复制节点。

[0070] 剩余复制节点的反应将在本文稍后讨论。如上所解释的,数据块可能具有比需要的更多的副本。未被选择为剩余复制节点的额外复制节点在写入锁被释放时不再是数据块的复制节点。即使在不涉及写入锁时,集群100的自治组件,如本文先前所讨论的,也可以自治地决定增加或减少数据块的最小和/或当前复制,如本文稍后所讨论的。

[0071] 在本文稍后讨论的实施例中,甚至在写入锁被释放之前,写入数据块可能导致其他节点处的数据块的副本无效。热图110中的脏列指示这样的无效。如图所示,节点C已写入节点123并将其标记为脏。

[0072] 但是,无效可能需要或可能不需要对读取锁的破坏。如本文稍后所讨论的,各种实施例可以急切地或懒惰地使数据块无效。例如,如图所示,节点B尚未注意到节点B的副本通过节点C的写入而无效和/或尚未对节点B的副本通过节点C的写入而无效做出反应。

[0073] 集群100的一致性模型可以提供写后读的语义,使得陈旧数据永远不被读取。例如,即使副本是分布式的并且不一定是集中管理的,集群100也可以提供符合POSIX的I/O,诸如写后读。如本文稍后所讨论的,写入器节点可以修改易失性存储器中的数据块,例如使用具有延迟刷新的I/O缓冲区,该延迟刷新可以包括持久化和复制。通过本文讨论的锁定行为,以及通过识别特定数据块或数据块系列的访问,多个读取器节点和多个写入器节点可以同时对符合POSIX的文件的不同相应部分操作,即使具有延迟的持久化和复制,并且不损害POSIX一致性。

[0074] 水平缩放的并行读取可能发生。例如,节点A可以:a)从节点B复制数据块123-124,b)替代地从节点C复制相同的块123-124,或c)同时从节点B复制数据块123以及从节点C复制数据块124。在实施例中,哪个复制节点提供数据块取决于哪个节点请求数据块,诸如使用本文较早讨论的分层网络。

[0075] 复制方式

[0076] 图2是描绘将数据块从一个节点复制到另一个节点的示例实现方式200的流程图。图2介绍了可能适用于本文稍后呈现的图的复制模式。图2中的各个组件可以是与图1或多或少相似的组件的实现。图2强调没有上下文的情况下的传送交互。复制的场合和场景将在本文稍后讨论。各种实施例可以具有各种传输机制,如本文稍后所讨论的。

[0077] 复制方式200包括以相应的步骤202A-202D开始的各种相互排斥的方式,其中一些共享后续步骤204和206,如图所示。为了说明的目的,图2需要包括如下各项的系统:a)数据块和热图,b)已经具有数据块的有效本地副本的发送方复制节点,c)正在成为复制节点或正在刷新数据块的其陈旧副本的接收方复制节点,以及d)如本文较早所解释的集群的自治组件。

[0078] 图2的各个步骤是由如下所示的那些各种组件执行的。如本文较早所解释的,复制主要由主动节点驱动,并且热图在大多数实施例中可以是被动数据结构。同样如本文较早

所解释的，复制有时可能需要集群的自治组件的发起和/或参与。如下所讨论的，步骤202C-202D和204需要自治组件的主动参与。如下所讨论，步骤202D实际上需要自治组件的自治活动。

[0079] 在步骤202A中，发送方复制节点直接将数据块推送给接收方复制节点。例如，发送方复制节点可以执行远程直接存储器访问(RDMA)写入以将数据块从发送方复制节点的易失性存储器复制到接收方复制节点的易失性存储器中。RDMA在本文稍后讨论。在各种实施例中并且尽管未示出，复制可能不完整直到：a)热图通过传送被更新，和/或b)接收器复制节点本地持久化数据块。

[0080] 在步骤202B中，发送方复制节点直接向接收方复制节点通知复制机会，诸如可能有条件或无条件需要复制的事件(occurrence)。如果接收方复制节点决定作出反应，则步骤206发生。在步骤206中，接收方复制节点诸如通过RDMA读取直接从发送方复制节点拉取数据块。在相关非专利文献(NPL)“Designing NFS With RDMA for Security,Performance and Scalability”中呈现了RDMA操作。

[0081] 在步骤202C中，发送方复制节点明确地或隐含地向自治组件通知复制机会，诸如可能有条件或无条件需要复制的事件。如果自治组件订阅、监视、观察或以其他方式接收或检测到发送方复制节点的活动(诸如对数据块的锁定、修改、持久化或解锁)的指示，则隐式通知可能发生。例如在各种实施例中，自治组件可以观察所示步骤中的一些或全部的发起和/或完成和/或步骤之间的所示过渡中的一些或全部。自治组件通过执行步骤204有条件地或无条件地作出反应。

[0082] 在步骤204中，自治组件向接收方复制节点通知复制机会，诸如可能有条件或无条件需要复制的事件。如果接收方复制节点决定作出反应，则步骤206如上所述发生。

[0083] 在步骤202D中，自治组件自治地激活自身，诸如在以下情况时激活自身：a)诸如用于冷却或重置访问计数的间隔计时器的周期性到期，b)节点、数据块或整个系统的某个性能指标阈值，和/或c)如根据贝叶斯网络的概率预测来指导的，如本文稍后所讨论的。自治组件可以自治地决定复制是需要的或期望的，在此情况下，步骤204和206可以如上所述发生。

[0084] 创建和丢弃副本

[0085] 图3描绘了随着各种场景的进展和交互可能发生以创建和丢弃数据块的副本的示例步骤序列301-310。图3是系统行为的宏观视图。如本文稍后呈现的，图4是可以出于各种原因接收和/或读取数据块的节点(诸如下面的读取器节点)的行为的微观视图。图3-图4可以描绘或可以不描绘相同的实施例。

[0086] 步骤301-310以所示顺序发生。然而，在一些条件或实施例下，给定步骤的子步骤可以在步骤内重新排序或并行化。例如，步骤309的子步骤A-B可以同时发生，如本文稍后讨论的。

[0087] 图3中的各种组件可以是图1中或多或少类似的组件的实现。为了说明的目的，图3需要包括如下各项的系统：a)数据块和热图，b)已经具有数据块的有效本地副本的原始复制节点，c)需要但缺少数据块的读取器节点和写入器节点，以及d)如本文较早所解释的集群的自治组件。

[0088] 图3的各种步骤由行动者列中所示的各种组件执行。相同步骤的子步骤由相同行

动者执行。在各种实施例中，自治组件可以观察一些或全部所示步骤和/或子步骤的发起和/或完成，即使自治组件当前是不活动的并且不以其他方式直接参与步骤。

[0089] 最初在步骤301中，原始复制节点如动作列中所示本地持久化数据块。如复制节点列所示，热图用于强制执行数据块的最小的一个副本，该副本驻留在原始复制节点上。换句话说，原始副本节点在易失性存储器和本地盘上有数据块的副本，并且没有其他节点包含数据块。

[0090] 在步骤302中，读取器节点自治复制并读取数据块。步骤302的未示出的子步骤可以包括：a)读取锁定数据块，b)诸如通过图2中呈现的某种方式使得数据块从原始复制节点被复制，c)将数据块存储在易失性和非易失性存储装置中，和/或d)诸如在由读取器节点的远程客户端的数据库查询执行期间，使用数据块的内容，诸如用于在线分析处理(OLAP)。在任何情况下，步骤302使读取器节点变成额外的复制节点，如复制节点列中所示，热图可以跟踪该复制节点。

[0091] 步骤303-304涉及节点的集群成员资格。在实施例中，集群具有仅在故障或维护期间波动的节点的静态清单。在实施例中，附加的节点可被添加以：a)通过提供更多未使用的盘空间来增加集群存储容量，b)通过在更多节点中的每一个节点上存储更少的副本来增加吞吐量，和/或c)通过增加最小副本来增加可靠性。

[0092] 在实施例中，集群具有可以单独并且自治地加入和离开集群的节点联合。示例联合包括：a)诸如在计算机云或数据网格中的弹性水平缩放，或)个人计算机、工作站和/或移动设备(诸如笔记本电脑和智能手机)的松散联合。

[0093] 在步骤303中，原始复制节点离开自治组件检测到的集群。例如，原始复制节点可以由于超时或心跳丢失(诸如当盘驱动器崩溃时)而明确离开或隐式离开。如果离开的节点是数据块的唯一复制节点，则这可能引起灾难性的数据丢失，这就是为什么更实际的示例永远不会将最小复制设置为低于2。

[0094] 自治组件对离开的节点如何作出反应可能如下变化。如果集群不再有数据块的最小副本，则不是复制节点的节点被选择成为另一个复制节点，以持久化丢失的副本的替换。当离开的节点使得多个数据块没有足够的副本时，那么一个或多个新的复制节点可能被需要。本文稍后讨论的是用于选择以下的启发式方法：a)招募多少附加的复制节点，b)那些附加的复制节点中的哪些应该接收哪些数据块，以及c)新的副本应该被从哪些幸存的复制节点复制。在一个示例中，一个新的复制节点接收所有替换副本。

[0095] 如步骤303的复制节点列中所示，读取器节点是数据块的唯一幸存的复制节点。因为当前最小的一个复制节点仍然可用，所以自治组件不需要招募附加的复制节点。

[0096] 在步骤304中，原始复制节点重新加入集群。在各种实施例或场景中，在重新加入后，一些或所有原始复制节点的本地副本清单丢失、陈旧或默认被忽略。在实施例中，集群不接受来自重新加入节点的副本。在所示实施例中，自治组件或重新加入的节点检测到重新加入的节点上的一些副本仍然有效，诸如当在离开和重新加入之间数据块在集群中未被修改时。在该情况下并且如步骤304的复制节点列中所示，集群再次具有比所需更多的数据块副本。

[0097] 在步骤305中，自治组件自治地决定将数据块的最小副本增加到二，如复制节点列中所示。因为数据块已经有两个副本，所以自治组件不需要招募另一个复制节点。否则，招

募将如上文所讨论的发生。

[0098] 步骤306-309涉及写入器节点，该写入器节点诸如用于诸如在线事务处理(OLTP)的数据库事务。在步骤306中，写入器节点对数据块写锁定。在子步骤306A中，并且因为写入器节点缺少并且需要数据块，所以从数据块的任何其他复制节点到写入器节点的数据块的复制诸如通过图2中所呈现的某种方式发生。用于选择从哪个复制节点复制的启发式方法在本文中稍后呈现并且可以由写入器节点决定。如步骤306的复制节点列所示，写入器节点成为另一个复制节点。

[0099] 在步骤306B中，写入锁定数据块导致该数据块上的所有读取锁被破坏，对此其他节点可以以各种方式和实施例作出反应，如本文稍后讨论的。但是，热图仍然跟踪哪些其他节点是复制节点，并且它们的副本尚未失效。例如，数据不会仅仅因为一个节点要写入锁定数据块并且然后立即崩溃而丢失。

[0100] 在步骤307中，写入器节点修改数据块。具体地，在子步骤307A中，写入器节点修改易失性存储器中的数据块。缓冲将在本文稍后讨论。

[0101] 在子步骤307B中，写入器节点将数据块标记为脏，这意味着被修改但未被本地持久化和/或未复制。在实施例中，热图记录哪些数据块是脏的。如本文稍后所讨论的，将数据块标记为脏可能最终导致懒惰的失效，诸如当复制节点最终且自治地检查以查看数据块是否已被标记为脏时。

[0102] 在步骤308中，当数据块被写入锁定时，任何其他节点可以尝试获取数据块的读取锁或写入锁。尝试被暂停直到当前写入锁被释放，并且尝试的节点等待所请求的锁被授予。

[0103] 在步骤309中，写入器节点通过解锁修正的数据块来提交和刷新它。在子步骤309A中，写入器节点选择最小数量的复制节点，包括写入器节点，以保留为复制节点。写入器节点的修正的副本被复制到剩余的复制节点，其在此情况下是写入器节点和原始复制节点，如子步骤309C的复制节点列中所示。实施例可以选择最新近的写入器节点、没有其他最近的写入器节点和最近的读取器节点作为剩余的复制节点。

[0104] 这样的复制可以通过图2中呈现的某种方式发生。超过最小复制的复制节点不再是数据块的复制节点，诸如如图所示的读取器节点。同样由解锁引起的是子步骤309B，其中写入器节点本地持久化修正的数据块。在子步骤309C中解锁完成，并且相同或不同的节点可以立即或最终锁定数据块以供不同的用途。

[0105] 步骤310需要自治组件的自治行为。在子步骤310A中，自治组件自治地减少数据块的最小复制，诸如在数据块的需求峰平静之后和/或在不同数据块的需求峰期间。在子步骤310B中，自治组件自治地取消数据块的多余复制节点。但是，取消多余的复制节点可以是可选的，并且在一些情况下不发生。

[0106] 因为写入器节点曾经最新近写入锁定了数据块，所以写入器节点仍然是复制节点。在此情况下，原始复制节点不再是数据块的复制节点。在另一个示例中，即使当最小复制量不变，数据块从不被写入锁定，并且集群成员资格不变时，自治组件也可以自治地取消多余的复制节点。例如，对数据或本地或系统性能指标的需求的动态波动可能引起自治组件的自治干预，该自治组件可以创建或丢弃相同或不同数据块的副本。

[0107] 读取生命周期

[0108] 图4描绘了读取器节点和数据块的示例读取生命周期400。在各种实施例中，一些

或所有所示步骤和/或所示步骤之间的一些或所有过渡的发起和/或完成可由自治组件观察和/或记录在热图中。所示步骤由读取器节点在以各种方式涉及相同数据块的各种场景期间执行。

[0109] 当相邻步骤或多或少以快速顺序或同时发生时，过渡被示为实线箭头。如果在无限长的持续时间后需要外部或异步刺激来引起过渡，则过渡被示为虚线箭头。

[0110] 在以该顺序执行步骤401-412的场景中，读取器节点缺少并且需要读取数据块。在步骤401中，读取器节点决定读取数据块，诸如当读取器节点的远程客户端在查询执行期间使用读取器节点进行数据检索时。如本文较早所讨论的，步骤402请求数据块的读取锁。

[0111] 请求读取锁的部分可能需要附加的活动，诸如可能在读取锁被授予之前发生的步骤403-404和415。步骤403和415被示为决策菱形，因为涉及请求读取锁的行为可能以数据块的局部性和/或数据块的脏度为条件。步骤403检测数据块是否已经驻留在读取器节点上。

[0112] 换句话说，步骤403检测读取器节点是否已经是数据块的复制节点，在此情况下读取可以被本地满足。如果数据块不是本地可用的，则步骤404潜在地等待数据块被写入器节点解锁。如果数据块已被解锁，则无需等待，并且步骤404直接完成。

[0113] 如果步骤403替代地检测到数据块已经本地可用，则步骤415检测数据块是否被写入器节点标记为脏。例如，当请求读取锁时，读取器节点可以检查当前标记为脏的数据块和/或数据块系列的全局目录。

[0114] 如果写入器节点已经解锁了数据块并且持久化并复制了数据块的修正，那么数据块未被写入锁定，并且未被列为脏的。因此，不脏或未被写入锁定的本地数据块的读取锁将在步骤405中被立即授予。因此，当数据块不脏时，步骤415之后紧接着是步骤405。

[0115] 而如果数据块本地可用但被写入器节点远程污染，则写入器节点尚未释放写入锁。在该情况下，步骤415之后是步骤404以等待写入锁的释放，如上文所解释的。在实施例中，从步骤415到404的过渡包括读取器节点通知并引起写入器节点刷新其缓冲的且脏的数据块，包括由写入器节点进行的复制和本地持久化。然而，这样的远程刷新不需要包括释放写入锁。

[0116] 如图所示，无论数据块是否是本地的，在步骤404中等待写入锁的释放都可能发生。因此，步骤404的等待可以在步骤403或415之后。最终释放写入锁，并在步骤405授予读取锁。在获取读取锁之后，读取器节点的后续行为取决于检测数据块是否已经本地可用的步骤406。

[0117] 步骤403和406的检测结果应该相同。如果数据块本地可用，则对数据块的直接使用可以在步骤411中发生。例如，读取器节点可以检查和分析数据块的内容。否则，数据块只是远程可用的并且应该被立即复制到读取器节点，这需要如下步骤407-410。

[0118] 步骤407使用RDMA读取，如本文较早针对图2所讨论的，以从另一节点将数据块的副本接收到读取器节点的易失性存储器中。一些实施例可能具有或可能不具有将数据块本地保存在易失性存储器中的特定位置处的步骤408。例如，最终，另一个节点可以使用RDMA读取以从易失性存储器中的该特定地址提取数据块。

[0119] 在实施例中，步骤407使用诸如操作系统(OS)输入/输出(I/O)缓冲区之类的缓冲区，该缓冲区保留在存储器中的特定地址处。在实施例中，步骤407使用诸如由数据库管理

系统(DBMS)管理的用于数据库块的缓冲区高速缓存中的缓冲区。在相关非专利文献(NPL)“Designing NFS With RDMA for Security,Performance and Scalability”中提出了RDMA缓冲和寻址。

[0120] 通过步骤407的易失性存储对于本文的复制可能是不够的,并且本地持久化接收到的数据块的附加步骤409也可能是必要的。例如,如果数据块是原始文件的一部分,则取决于实施例和场景,数据块可被:a)本地持久化到原始文件的整体或部分本地副本内的对应块位置中,b)自身本地持久化为一个单块文件,或d)本地持久化到无序数据块的另一个文件中,这些数据块被编目以供随机访问。例如,各种数据块最初可以不考虑原始文件内的排序而被持久化,并且稍后被重新组合为原始文件的连续部分或全部。

[0121] 在步骤410中,读取器节点在本地持久化数据块之后在热图中明确记录复制。通过在步骤410中持久化本地副本,读取器节点已经成为该数据块的复制节点,并且在步骤411中读取器节点可以直接使用本地副本的内容。因此,无论是复制被需要还是数据块已经在本地可用,步骤411都可以发生。步骤411的前提条件可以是数据块驻留在易失性和非易失性本地存储装置中。

[0122] 在步骤411之后,读取器节点对数据块的读取和使用可以完成,但是读取器节点在步骤412中仍然是数据块的复制节点。在各种实施例中,读取器节点在完成使用数据块时释放或不释放数据块上的其读取锁。因为读取器节点仍然是数据块的复制节点,所以步骤413可以从另一个节点接收读取请求,当步骤414诸如通过RDMA向易失性存储器发送数据块的副本和从易失性存储器发送数据块的副本时该另一个节点也变成复制节点,之后步骤412的就绪状态被再次访问。因此,读取器节点可以在或多或少地无限期地处于就绪状态时保留和提供数据块的副本。

[0123] 当处于步骤412的就绪状态时并且虽然未示出,自治组件或写入器节点可以取消读取器节点作为数据块的复制节点。在该情况下,读取器节点可以从易失性和非易失性存储器中丢弃数据块。

[0124] 当处于步骤412的就绪状态时,读取器节点可以检测到或被通知数据块的本地副本是陈旧的,因为写入器节点将数据块标记为脏或提交了数据块的修正。在该情况下以及在各种实施例中,无效步骤417发生,其可以从易失性和非易失性存储器中丢弃数据块,并且在一些情况下,导致数据块被再次复制到读取器节点。

[0125] 即使当处于步骤412的就绪状态时,读取器节点对稍后使用数据块的任何尝试都应该在步骤401再次开始,包括重新锁定。当步骤401-403、415、405-406和411以该顺序被重复时,数据块的本地副本可以在本地重复使用而无需复制。否则,需要如上所述重复复制。

[0126] 如本文较早所讨论的,自治组件可以主动且自治地行动以引起节点未请求的复制。例如,复制节点可能崩溃,或者自治组件可能自治地提高数据块的最小副本量。在这样的情况下,自治组件可以自治招募节点以成为复制节点。例如,如果读取器节点还不是数据块的复制节点,则自治组件可以在步骤416中使读取器节点变成复制节点,其中复制如上所述发生。

[0127] 用于复制的示例自治和/或热图使用

[0128] 图5描绘了其中复制和热图可以相互影响的示例计算机过程。涉及的系统可能包括数据块、热图、自治组件和节点,包括请求节点。

[0129] 尽管步骤501是准备性的并且需要在各种节点上持久化数据块的副本，但是步骤501可以明确地作为持久数据的初始分布和存储或通过如本文较早讨论的较早复制的操作而发生。例如，各种文件可以包含各种数据块，并且文件的全部或部分副本可以分布并存储在各种节点的本地盘中。

[0130] 在步骤502中，请求节点请求对各种(一个或多个)数据块(诸如一系列数据块)进行读取或写入访问。如较早所讨论的，步骤502可能需要在涉及的数据块上获取读取锁或写入锁。

[0131] 基于所请求的访问，在步骤503修改热图。例如，读取计数器或写入计数器可以分别与被访问的每个数据块或数据块系列相关联。根据本文较早提出的技术向热图通知访问。例如，请求或授予锁可以导致递增热图。

[0132] 基于所请求的访问，步骤504复制一些或所有被访问的数据块以及可能的其他数据块。例如，对相同文件的许多连续数据块的顺序访问可以：a)重用一些数据块的本地副本，b)导致对相同访问请求所需的其他数据块的复制，和/或c)主动(即急切地和自治地)复制可能很快被需要的相同文件的未请求的数据块。例如，步骤504可以提前读取以预取一些数据块，诸如用于表扫描。

[0133] 基于热图，在步骤505中针对至少一个数据块、数据块系列或文件调整复制节点的最小数量。如本文较早所讨论的，增加或减少副本的最小数量可以响应于自治组件的自治和主动决策而发生。自治组件可以分别在数据块的需求峰开始和结束处增加然后减少数据块的最小数量。如下所述，自治组件可以将一些决策委托给贝叶斯网络，该贝叶斯网络基于如从(诸如热图中观察和/或记录的)数据块和节点的历史访问中学习到的相应可识别模式发生的概率来预测最佳配置和调整。

[0134] 复制模式

[0135] 用于自动副本优化的逻辑的各种实施例可以对整个集群或节点、数据块和/或文件的相应子集施加各种复制模式，诸如以下复制模式：

[0136] A.全N路——此类型的复制在N个节点之间复制数据，等待所有写入完成，然后再返回到发起客户端。这是最慢的修改类型。

[0137] B.故障转移组——类似于N路，但节点被组织成节点组，并且数据和修正必须复制到每个组中的至少一个节点。这有助于根据用户的需要(可能跨机架或其他特性)定义数据冗余区域。

[0138] C.单一——没有复制。

[0139] D.异步——数据和修正跨节点复制，无需等待其他节点确认请求。这是最快的复制方式，但不提供数据被实际复制的完美的保证。

[0140] E.本地镜像+N远程——本地盘或文件系统可被配置为本地副本。这提供了针对盘而非节点故障的冗余。可以使用第二远程节点(或N个远程节点)来防护节点故障，这确保数据在主节点丢失后仍然幸存。

[0141] F.本地镜像+N远程异步——本地盘或文件系统可被配置为本地副本——这提供了针对盘但非节点故障的冗余。可以使用第二远程节点(或N个远程节点)来防护节点故障，这确保数据在主节点丢失的情况下持久化。远程副本是异步的，以提供快速的本地冗余。

[0142] G.机器学习和/或持续调优——通过使用跟踪机制，文件系统访问模式可被学习

和识别。热图可被查阅以便以适合或不适合其他严格复制模式的方式跨集群分布数据。结合至少一种其他复制模式，数据可以被放置在其被频繁访问的区域中，从而确保未来访问的更低延迟。

[0143] 因此，自治组件可以施加复制模式和在复制模式之间切换。在复制模式被施加后，相同或不同的自治组件可以在持续的基础上进一步适应和调优复制配置。

[0144] 网络传输

[0145] 各种实施例可以使用各种网络传输来促进复制，诸如RDMA。在实施例中，使用或不使用RDMA的传送可能需要使用无连接或未确认的网络传输(诸如用户数据报协议(UDP))以用于增加的吞吐量。例如，无限带宽具有支持和/或加速UDP传输的各种传输模式。UDP还邀请特定于应用程序的数据包排序和重传，诸如当丢失的数据包可能有条件地需要重新发送时。

[0146] 尽管节点的存储集群在传输层处逻辑上是全连接的，传输层是开放系统互连(OSI)网络通信模型中的层堆栈的一层，但在传输层下方的网络层处，集群的网络拓扑可能带来不对称性，诸如(诸如对于互联网络的或对于存储转发多跳距离的)不同的通信链路带宽以及可能导致瓶颈的不同链路利用率。通信结构可能包含网络交换机的层次结构。与网络路由中的因素和关注点类似的因素和关注点可以被包括在自动副本优化逻辑的决策中。这样的决策可以(诸如来自无限带宽拓扑的)集成静态指标和事实以及动态指标和事实(诸如链路和/或节点的相对利用率以及松散联合集群中的节点成员资格)。

[0147] 传输可以诸如通过不受信任的互联网络被加密。网络会话在节点加入集群时开始，在节点离开集群时结束，无论传输是有连接还是无连接。网络会话可能需要选择特定数据块用于传送，诸如数据库块。

[0148] 例如，仅不连续的数据块可被发送以用于随机访问。诸如当：a)表扫描时，或b)发送文件的全部或部分时，诸如当文件系统用于持久化时，比所请求更多的数据块的序列可被发送。

[0149] 各种实施例可以使用针对低延迟传输优化的各种通信链路技术，诸如光纤通道(FC)、互联网小型计算机系统接口(iSCSI)和以太网光纤通道(FCoE)。各种实施例可以使用各种RDMA协议来避免诸如来自盘旋转或轨道切换的机械驱动延迟。例如，RDMA可以将副本递送到读取器节点的易失性存储器中，然后以下可能同时发生：a)读取器节点将易失性内容用于特定于应用程序的目的，以及b)内容被本地持久化。RDMA协议包括融合以太网上的RDMA(RoCE)和iWARP。

[0150] 贝叶斯网络

[0151] 在一些程度上，节点集群可以作为分布式高速缓存操作，其延迟、吞吐量、能耗和盘驱动器寿命可能取决于数据块分布模式。例如，OLTP延迟可能取决于副本的时间和空间局部性。诸如在诸如由OLAP进行的表扫描期间，许多数据块的顺序访问可能使所需副本的放置中断。写入后对复制节点的选择可能危及副本的现有放置。

[0152] 自动副本优化逻辑可以包括贝叶斯网络或由贝叶斯网络指导，该贝叶斯网络基于过去的访问和学习的未来访问概率来预测各种最佳副本分布。各种贝叶斯网络可以计算相应事件的概率，诸如以下的可能性：

[0153] • 两个数据块将被一起访问，

[0154] • 一个数据块将被访问，因为另一个数据块已被访问，

[0155] • 特定节点将访问特定数据块，或者

[0156] • 锁将在持续时间内释放。

[0157] 例如，贝叶斯网络可能会学习到一些节点或应用程序：持有锁比其他节点或应用程序时间更长，或者具有或多或少稳定的数据块工作集。

[0158] 硬件概述

[0159] 根据一个实施例，本文描述的技术由一个或多个专用计算设备实现。专用计算设备可以是硬连线的以执行这些技术，或者可以包括数字电子设备(诸如被持久地编程以执行这些技术的一个或多个专用集成电路(ASIC)或现场可编程门阵列(FPGA))，或者可以包括被编程为根据固件、存储器、其它存储装置或组合中的程序指令执行这些技术的一个或多个通用硬件处理器。这种专用计算设备还可以将定制的硬连线逻辑、ASIC或FPGA与定制编程相结合，以实现这些技术。专用计算设备可以是台式计算机系统、便携式计算机系统、手持设备、联网设备或者结合硬连线和/或程序逻辑以实现这些技术的任何其它设备。

[0160] 例如，图6是图示可以在其上实现本发明实施例的计算机系统600的框图。计算机系统600包括总线602或用于传送信息的其它通信机构，以及与总线602耦合以处理信息的硬件处理器604。硬件处理器604可以是例如通用微处理器。

[0161] 计算机系统600还包括耦合到总线602的主存储器606，诸如随机存取存储器(RAM)或其它动态存储设备，用于存储将由处理器604执行的信息和指令。主存储器606还可以用于在执行将由处理器604执行的指令期间存储临时变量或其它中间信息。当存储在处理器604可访问的非暂时性存储介质中时，这些指令使计算机系统600成为被定制以执行指令中指定的操作的专用机器。

[0162] 计算机系统600还包括耦合到总线602的只读存储器(ROM)608或其它静态存储设备，用于存储用于处理器604的静态信息和指令。存储设备610(诸如磁盘、光盘或固态驱动器)被提供并耦合到总线602，用于存储信息和指令。

[0163] 计算机系统600可以经由总线602耦合到显示器612(诸如阴极射线管(CRT))，用于向计算机用户显示信息。包括字母数字键和其它键的输入设备614耦合到总线602，用于将信息和命令选择传送到处理器604。另一种类型的用户输入设备是光标控件616(诸如鼠标、轨迹球或光标方向键)，用于将方向信息和命令选择传送到处理器604并用于控制显示器612上的光标移动。这种输入设备通常在两个轴(第一轴(例如，x)和第二轴(例如，y))上具有两个自由度，这允许设备指定平面中的位置。

[0164] 计算机系统600可以使用定制的硬连线逻辑、一个或多个ASIC或FPGA、固件和/或程序逻辑(它们与计算机系统相结合，使计算机系统600成为或将计算机系统600编程为专用机器)来实现本文所述的技术。根据一个实施例，响应于处理器604执行包含在主存储器606中的一个或多个指令的一个或多个序列，计算机系统600执行本文所述的技术。这些指令可以从另一个存储介质(诸如存储设备610)读入到主存储器606中。包含在主存储器606中的指令序列的执行使得处理器604执行本文所述的处理步骤。在替代实施例中，可以使用硬连线的电路系统代替软件指令或与软件指令组合。

[0165] 如本文使用的术语“存储介质”是指存储使机器以特定方式操作的数据和/或指令的任何非暂时性介质。这种存储介质可以包括非易失性介质和/或易失性介质。非易失性介

质包括例如光盘、磁盘或固态驱动器，诸如存储设备610。易失性介质包括动态存储器，诸如主存储器606。存储介质的常见形式包括例如软盘、柔性盘、硬盘、固态驱动器、磁带或任何其它磁数据存储介质、CD-ROM、任何其它光学数据存储介质、任何具有孔图案的物理介质、RAM、PROM和EPROM、FLASH-EPROM、NVRAM、任何其它存储器芯片或盒式磁带。

[0166] 存储介质不同于传输介质但可以与传输介质结合使用。传输介质参与在存储介质之间传送信息。例如，传输介质包括同轴电缆、铜线和光纤，包括包含总线602的导线。传输介质也可以采用声波或光波的形式，诸如在无线电波和红外数据通信期间生成的那些。

[0167] 将一个或多个指令的一个或多个序列携带到处理器604以供执行可以涉及各种形式的介质。例如，指令最初可以在远程计算机的磁盘或固态驱动器上携带。远程计算机可以将指令加载到其动态存储器中，并使用调制解调器通过电话线发送指令。计算机系统600本地的调制解调器可以在电话线上接收数据并使用红外发送器将数据转换成红外信号。红外检测器可以接收红外信号中携带的数据，并且适当的电路系统可以将数据放在总线602上。总线602将数据携带到主存储器606，处理器604从主存储器606检索并执行指令。由主存储器606接收的指令可以可选地在由处理器604执行之前或之后存储在存储设备610上。

[0168] 计算机系统600还包括耦合到总线602的通信接口618。通信接口618提供耦合到网络链路620的双向数据通信，其中网络链路620连接到本地网络622。例如，通信接口618可以是集成服务数字网(ISDN)卡、电缆调制解调器、卫星调制解调器或者提供与对应类型的电话线的数据通信连接的调制解调器。作为另一个示例，通信接口618可以是局域网(LAN)卡，以提供与兼容LAN的数据通信连接。还可以实现无线链路。在任何此类实现中，通信接口618都发送和接收携带表示各种类型信息的数字数据流的电信号、电磁信号或光信号。

[0169] 网络链路620通常通过一个或多个网络向其它数据设备提供数据通信。例如，网络链路620可以提供通过本地网络622到主计算机624或到由互联网服务提供商(ISP)626操作的数据设备的连接。ISP 626进而通过全球分组数据通信网络(现在通常称为“互联网”628)提供数据通信服务。本地网络622和互联网628都使用携带数字数据流的电信号、电磁信号或光信号。通过各种网络的信号以及在网络链路620上并通过通信接口618的信号(其将数字数据携带到计算机系统600和从计算机系统600携带数字数据)是传输介质的示例形式。

[0170] 计算机系统600可以通过(一个或多个)网络、网络链路620和通信接口618发送消息和接收数据，包括程序代码。在互联网示例中，服务器630可以通过互联网628、ISP 626、本地网络622和通信接口618发送对应用程序的所请求代码。

[0171] 接收到的代码可以在它被接收到时由处理器604执行，和/或存储在存储设备610或其它非易失性存储装置中以供稍后执行。

[0172] 软件概述

[0173] 图7是可以用于控制计算系统600的操作的基本软件系统700的框图。软件系统700及其组件，包括它们的连接、关系和功能，仅仅意在是示例性的，并且不意味着限制(一个或多个)示例实施例的实现。适于实现(一个或多个)示例实施例的其它软件系统可以具有不同的组件，包括具有不同的连接、关系和功能的组件。

[0174] 软件系统700被提供以用于指导计算系统600的操作。可以存储在系统存储器(RAM)606和固定存储装置(例如，硬盘或闪存)610上的软件系统700包括内核或操作系统(OS)710。

[0175] OS 710管理计算机操作的低级方面,包括管理进程的执行、存储器分配、文件输入和输出(I/O)以及设备I/O。表示为702A、702B、702C...702N的一个或多个应用程序可以被“加载”(例如,从固定存储装置610传送到存储器606中)以供系统700执行。意图在计算机系统600上使用的应用或其它软件也可以被存储为可下载的计算机可执行指令集,例如,用于从互联网位置(例如,Web服务器、app商店或其它在线服务)下载和安装。

[0176] 软件系统700包括图形用户界面(GUI)715,用于以图形(例如,“点击”或“触摸手势”)方式接收用户命令和数据。进而,可以由系统700根据来自操作系统710和/或(一个或多个)应用702的指令来作用于这些输入。GUI 715还用于显示来自OS 710和(一个或多个)应用702的操作结果,用户可以对其提供附加的输入或终止会话(例如,注销)。

[0177] OS 710可以直接在计算机系统600的裸硬件720(例如,(一个或多个)处理器604)上执行。可替代地,管理程序或虚拟机监视器(VMM)730可以插入在裸硬件720和OS 710之间。在这个配置中,VMM 730充当OS 710与计算机系统600的裸硬件720之间的软件“缓冲”或虚拟化层。

[0178] VMM 730实例化并运行一个或多个虚拟机实例(“访客机”)。每个访客机包括“访客”操作系统(诸如OS 710),以及被设计为在访客操作系统上执行的一个或多个应用(诸如(一个或多个)应用702)。VMM 730向访客操作系统呈现虚拟操作平台并管理访客操作系统的执行。

[0179] 在一些情况下,VMM 730可以允许访客操作系统如同其直接在计算机系统700的裸硬件720上运行一样运行。在这些实例中,被配置为直接在裸硬件720上执行的访客操作系统的相同版本也可以在VMM 730上执行而无需修改或重新配置。换句话说,VMM 730可以在一些情况下向访客操作系统提供完全硬件和CPU虚拟化。

[0180] 在其它情况下,访客操作系统可以被专门设计或配置为在VMM730上执行以提高效率。在这些实例中,访客操作系统“意识到”它在虚拟机监视器上执行。换句话说,VMM 730可以在一些情况下向访客操作系统提供半虚拟化。

[0181] 计算机系统进程包括硬件处理器时间的分配以及(物理和/或虚拟)存储器的分配,存储器的分配用于存储由硬件处理器执行的指令、用于存储由硬件处理器执行指令所生成的数据、和/或用于当计算机系统进程未运行时在硬件处理器时间的分配之间存储硬件处理器状态(例如,寄存器的内容)。计算机系统进程在操作系统的控制下运行,并且可以在计算机系统上执行的其它程序的控制下运行。

[0182] 云计算

[0183] 本文一般地使用术语“云计算”来描述计算模型,该计算模型使得能够按需访问计算资源(诸如计算机网络、服务器、软件应用和服务)的共享池,并且允许以最少的管理工作或服务提供商交互来快速提供和释放资源。

[0184] 云计算环境(有时称为云环境或云)可以以各种不同方式实现,以最好地适应不同要求。例如,在公共云环境中,底层计算基础设施由组织拥有,该组织使其云服务可供其它组织或公众使用。相反,私有云环境一般仅供单个组织使用或在单个组织内使用。社区云旨在由社区内的若干组织共享;而混合云包括通过数据和应用可移植性绑定在一起的两种或更多种类型的云(例如,私有、社区或公共)。

[0185] 一般而言,云计算模型使得先前可能由组织自己的信息技术部门提供的那些职责

中的一些代替地作为云环境内的服务层来交付，以供(根据云的公共/私人性质，在组织内部或外部的)消费者使用。取决于特定实现，由每个云服务层提供或在每个云服务层内提供的组件或特征的精确定义可以有所不同，但常见示例包括：软件即服务(SaaS)，其中消费者使用在云基础设施上运行的软件应用，同时SaaS提供者管理或控制底层云基础设施和应用。平台即服务(PaaS)，其中消费者可以使用由PaaS提供者支持的软件编程语言和开发工具，以开发、部署和以其它方式控制它们自己的应用，同时PaaS提供者管理或控制云环境的其它方面(即，运行时执行环境下的一切)。基础设施即服务(IaaS)，其中消费者可以部署和运行任意软件应用，和/或提供处理、存储、网络和其它基础计算资源，同时IaaS提供者管理或控制底层物理云基础设施(即，操作系统层下面的一切)。数据库即服务(DBaaS)，其中消费者使用在云基础设施上运行的数据库服务器或数据库管理系统，同时DbaaS提供者管理或控制底层云基础设施和应用。

[0186] 为了说明可以用于实现(一个或多个)示例实施例的基本底层计算机组件而呈现了上述基本计算机硬件和软件以及云计算环境。但是，(一个或多个)示例实施例不必限于任何特定的计算环境或计算设备配置。代替地，根据本公开，(一个或多个)示例实施例可以在本领域技术人员将理解为能够支持本文呈现的(一个或多个)示例实施例的特征和功能的任何类型的系统体系架构或处理环境中实现。

[0187] 在前面的说明书中，已经参考众多具体细节描述了本发明的实施例，这些细节可以根据实施方式有所变化。因而，说明书和附图应被视为说明性而非限制性的。本发明范围的唯一和排他性指示，以及申请人意图作为本发明范围的内容，是以发布这种权利要求的具体形式从本申请发布的权利要求集合的字面和等同范围，包括任何后续更正。

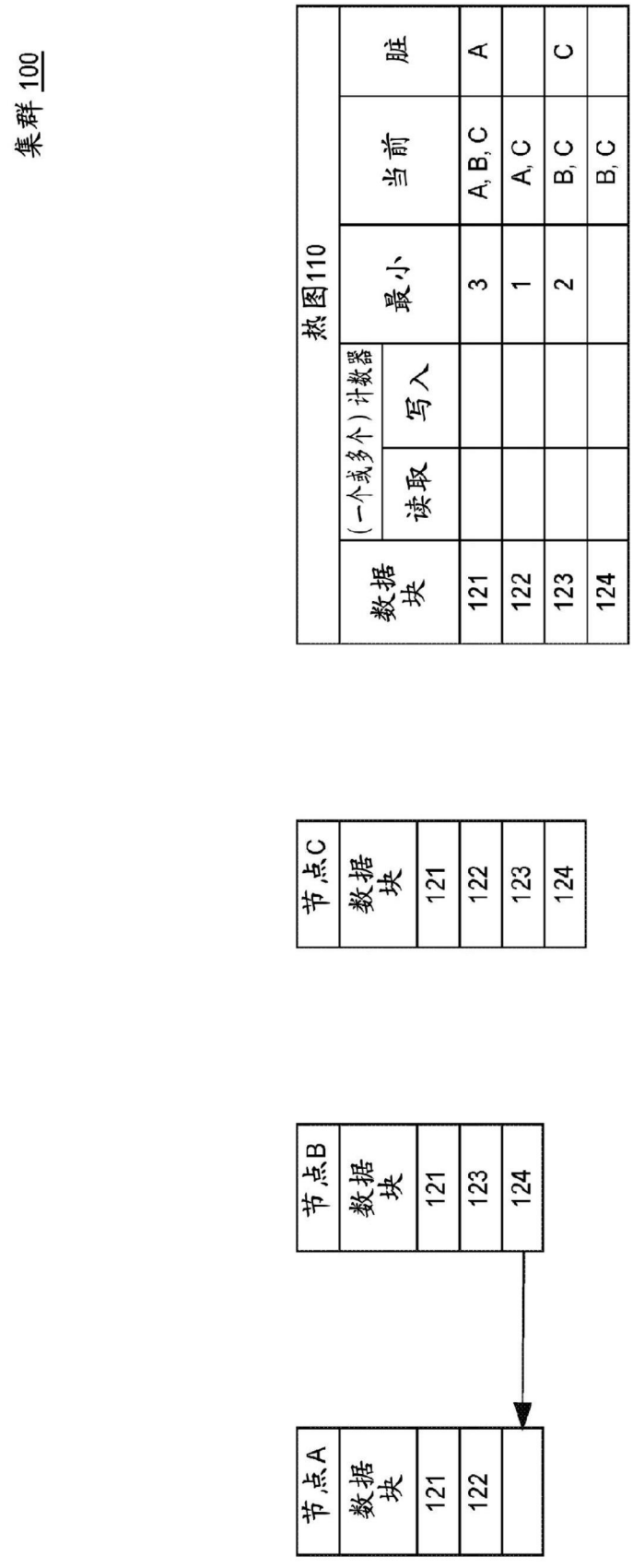

集群 100
热图110
数据块
(一个或多个)计数器
读取
写入
最小
当前
脏
121
3
A,B,C
A
122
1
A,C
123
2
B,C
C
124
B,C
节点C
数据块
121
122
123
124
节点B
数据块
121
123
124
节点A
数据块
121
122

图1

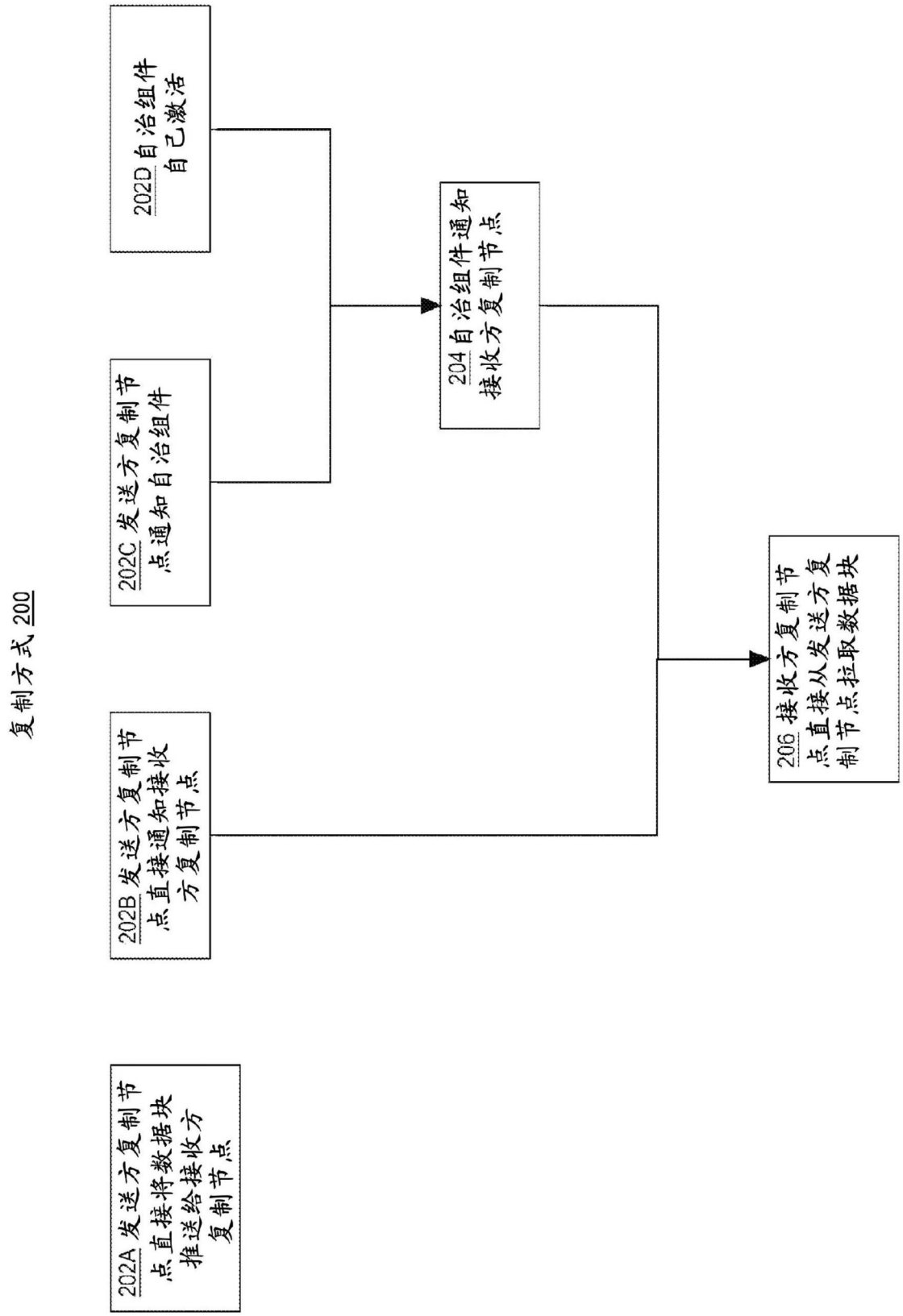

复制方式 200
202A 发送方复制节点直接将数据块推送给接收方复制节点
202B 发送方复制节点直接通知接收方复制节点
202C 发送方复制节点通知自治组件
202D 自治组件自己激活
204 自治组件通知接收方复制节点
206 接收方复制节点直接从发送方复制节点拉取数据块

图2

步骤	子步骤	行动者	动作	复制节点
301		原始复制节点	本地持久化数据块	最小=1， 原始复制节点
302		读取器节点	提取及查看数据块	原始复制节点、 读取器节点
303		自治组件	检测到原始复制节点离开集群	读取器节点
304		原始复制节点	重新加入集群	原始复制节点、 读取器节点
305		自治组件	将数据块的最小复制增加到二	最小=2
306	A	写入器节点	从一些复制节点提取数据块	原始复制节点、 读取器节点、 写入器节点
	B		打破数据块的读取锁	
307	A		修改易失性存储器中的数据块	
	B		将数据块标记为脏	
308		任何节点	等待锁定以及读取或写入数据块	
309	A	写入器节点	将修正的数据块推送到最小复制节点	
	B		本地持久化修正的数据块	
	C		解锁数据块	原始复制节点、 写入器节点
310	A	自治组件	将数据块的最小复制减少到一	最小=1
	B		取消数据块的多余复制节点	写入器节点

时间

图3

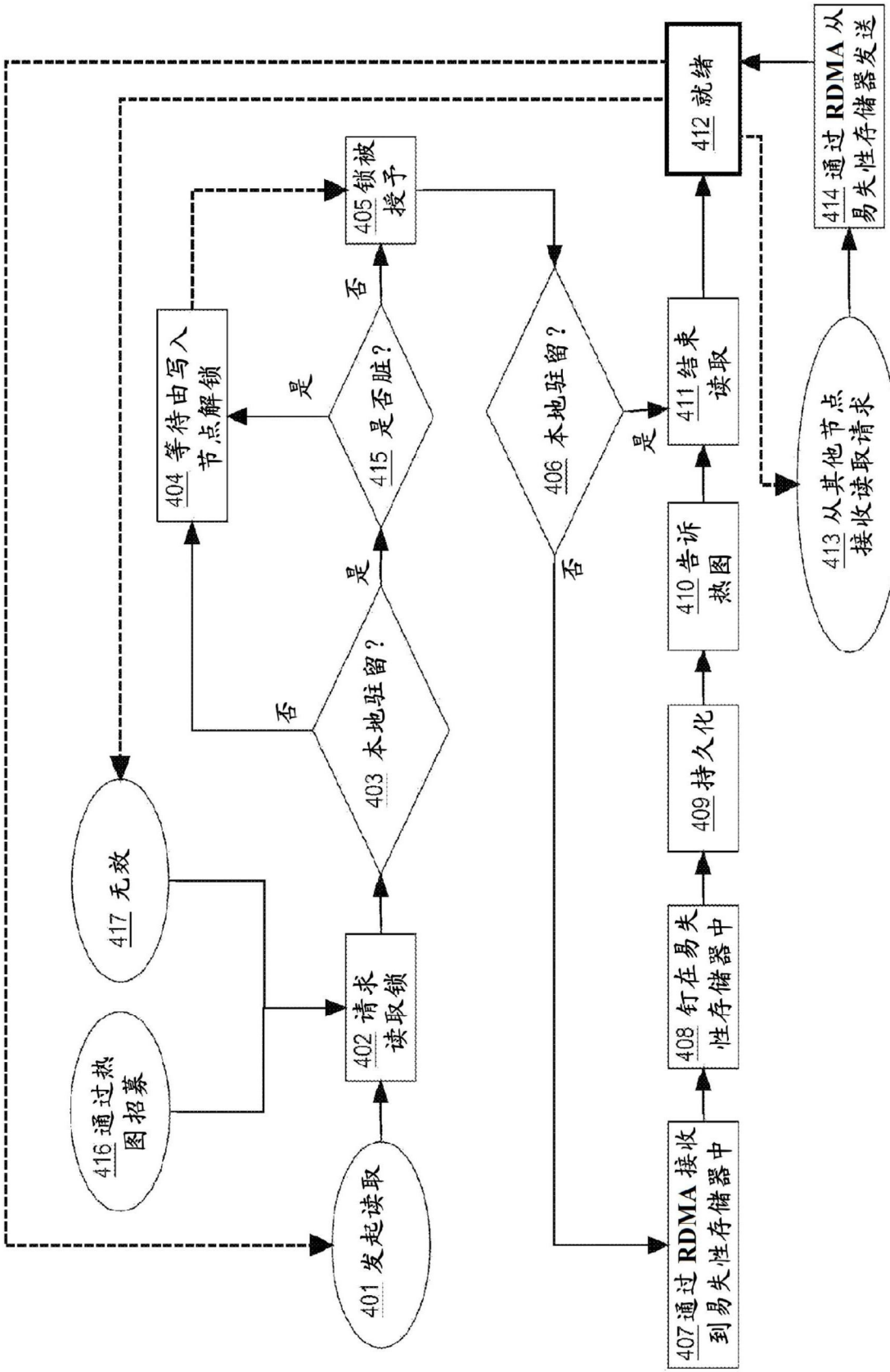

401 发起读取
402 请求读取锁
403 本地驻留?
404 等待由写入节点解锁
405 锁被授予
406 本地驻留?
407 通过 RDMA 接收到易失性存储器中
408 钉在易失性存储器中
409 持久化
410 告诉热图
411 结束读取
412 就绪
413 从其他节点接收读取请求
414 通过 RDMA 从易失性存储器发送
415 是否脏?
416 通过热图招募
417 无效
是
否
读取生命周期 400

图4

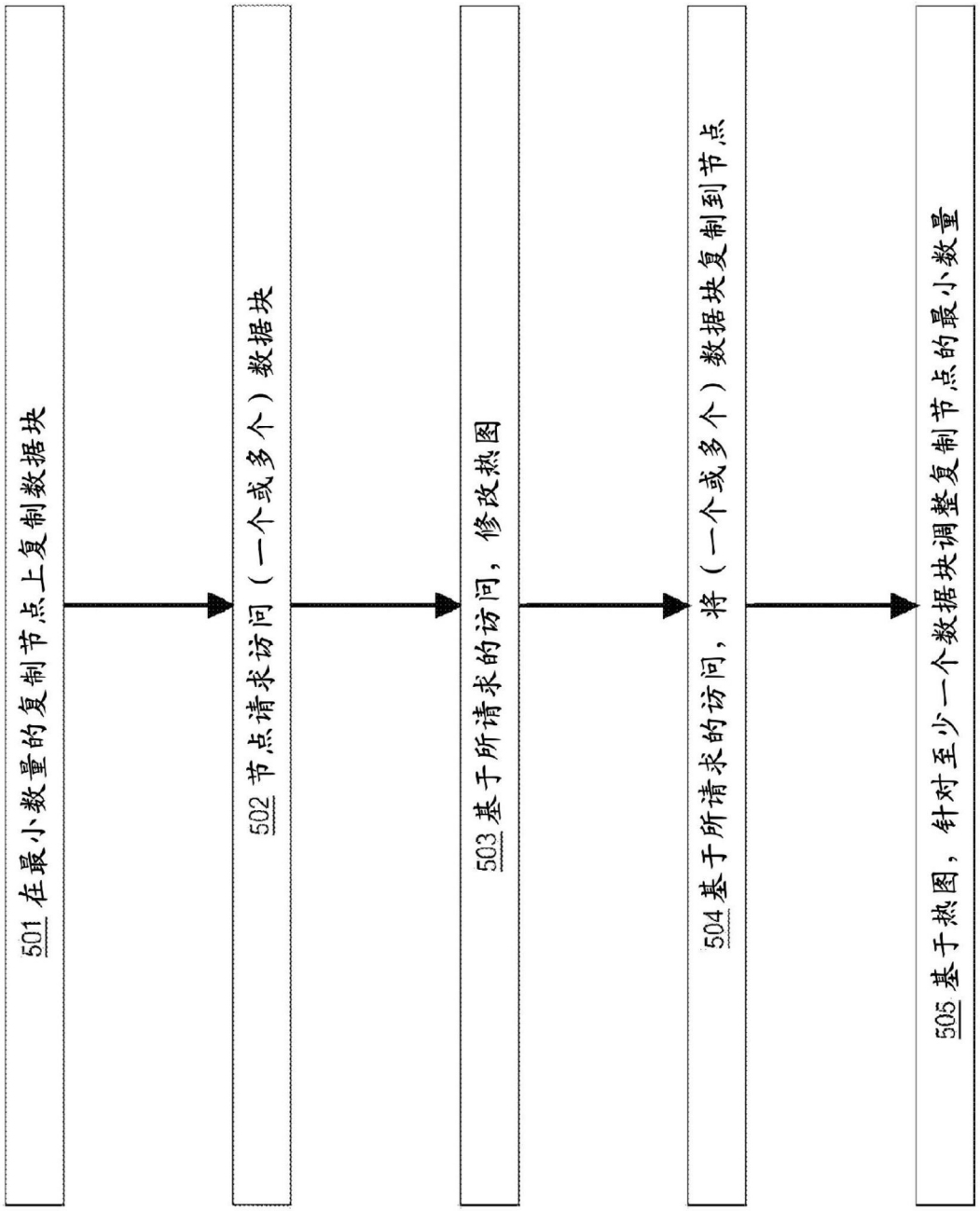

501 在最小数量的复制节点上复制数据块
502 节点请求访问（一个或多个）数据块
503 基于所请求的访问，修改热图
504 基于所请求的访问，将（一个或多个）数据块复制到节点
505 基于热图，针对至少一个数据块调整复制节点的最小数量

图5

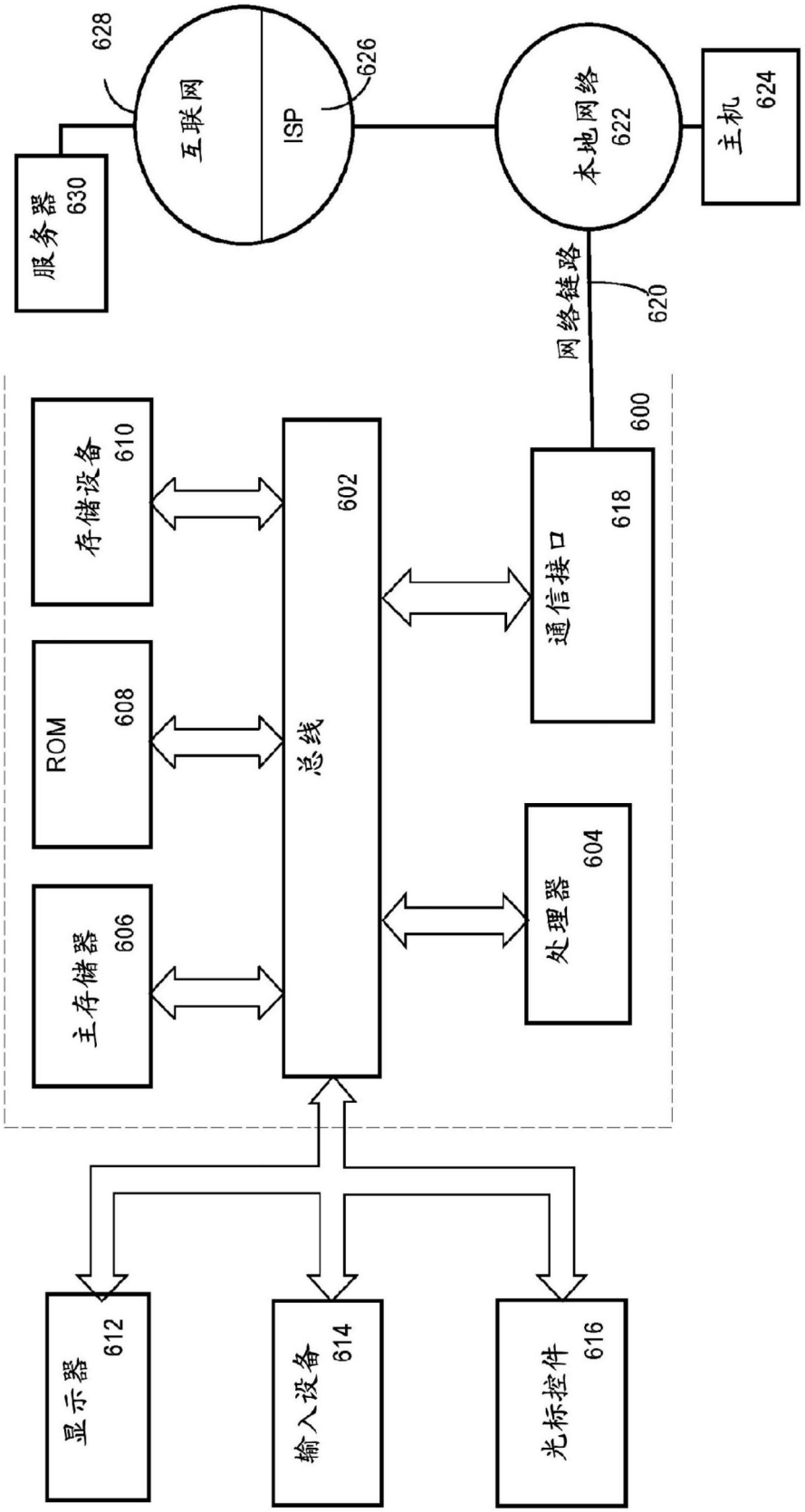

628
互联网
ISP
626
本地网络
622
主机
624
服务器
630
网络链路
620
600
存储设备
610
602
通信接口
618
ROM
608
总线
主存储器
606
处理器
604
显示器
612
输入设备
614
光标控件
616

图6

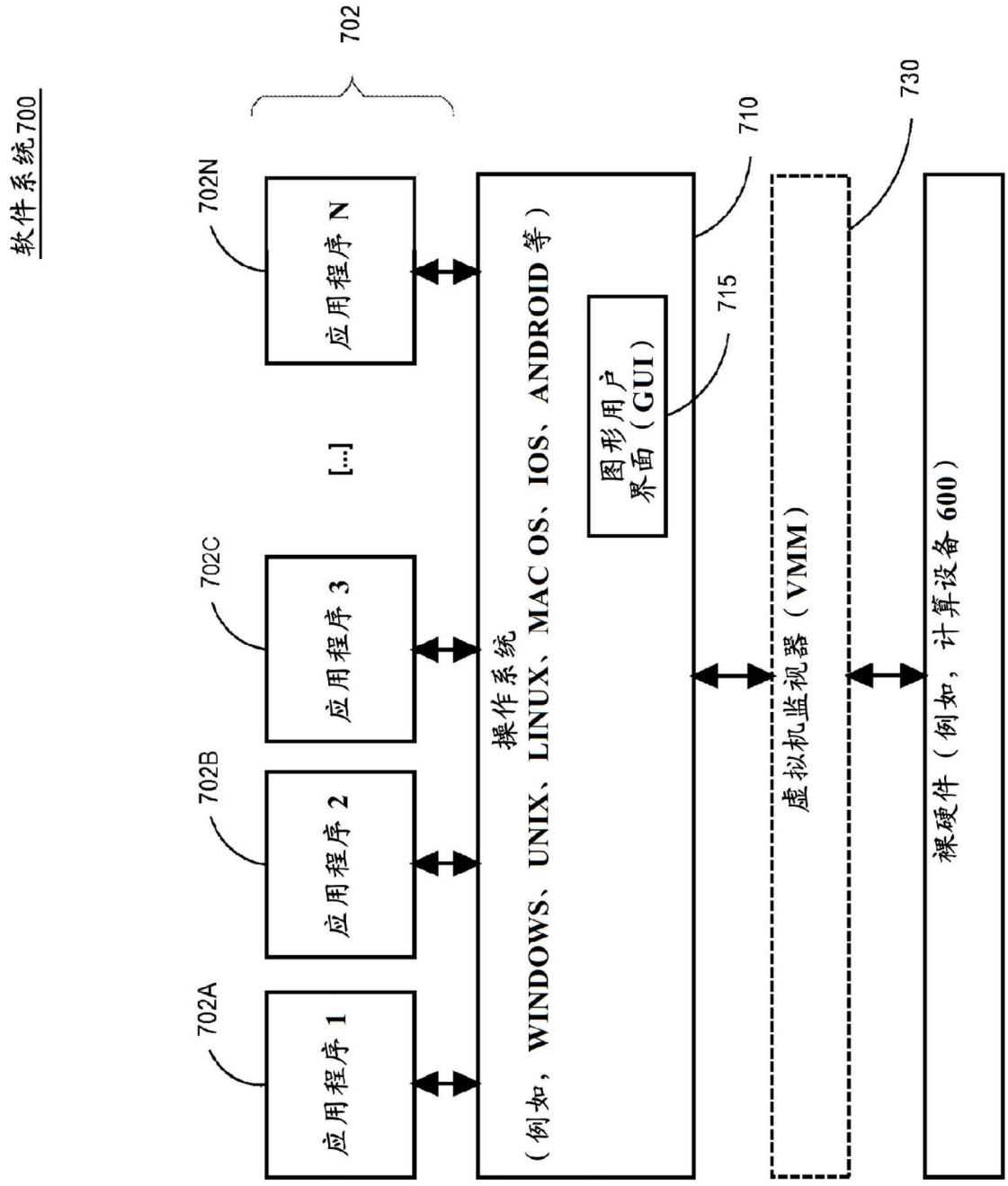

软件系统700
702
702A
应用程序 1
702B
应用程序 2
702C
应用程序 3
[...]
702N
应用程序 N
操作系统
（例如，WINDOWS、UNIX、LINUX、MAC OS、IOS、ANDROID 等）
710
图形用户界面（GUI）
715
虚拟机监视器（VMM）
730
裸硬件（例如，计算设备 600）

图7