



(12) 发明专利申请

(10) 申请公布号 CN 102549524 A

(43) 申请公布日 2012.07.04

(21) 申请号 201080043017.5

(74) 专利代理机构 中国专利代理(香港)有限公司 72001

(22) 申请日 2010.07.19

代理人 殷霖晨 王忠忠

(30) 优先权数据

61/226774 2009.07.20 US

(51) Int. Cl.

G06F 1/32 (2006.01)

(85) PCT申请进入国家阶段日

G06F 15/16 (2006.01)

2012.03.20

(86) PCT申请的申请数据

PCT/US2010/042478 2010.07.19

(87) PCT申请的公布数据

W02011/011336 EN 2011.01.27

(71) 申请人 卡林戈公司

地址 美国德克萨斯州

(72) 发明人 J·E·杜顿 L·阿比拉

D·约亚克利

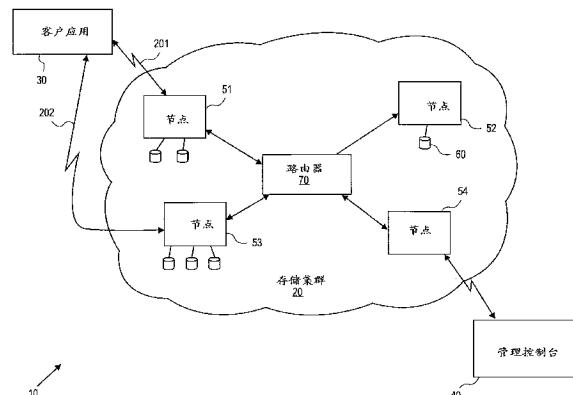
权利要求书 5 页 说明书 12 页 附图 5 页

(54) 发明名称

存储集群中的自适应功率保存

(57) 摘要

固定内容存储集群中的每个节点和卷基于在可配置时期内没有收到来自客户应用和节点的请求而作出关于是否减少功耗的独立决定。节点配置参数 sleepAfter 和 wakeAfter 分别确定在使节点或卷闲置之前需要等待多久以及在重新执行完整性检查之前需要在闲置时等待多久。竞标值由每个节点计算，该竞标值反映节点写入文件、读取文件或者保持文件的拷贝的成本是多少。具有最低竞标值的节点获胜，并且闲置的节点向每个竞标添加补贴，以帮助确保闲置节点是保持闲置的。通常，具有较大容量的节点将提交较低的竞标值以写入文件。在存档模式下，写入竞标被改变，这意味着具有较小容量的节点提交较低的竞标值，从而意味着较满的节点更快地充满，并且然后闲置，而空的或几乎空的节点可以保持闲置一定时间，直至赢得写入竞标。



1. 在相互连接的多个计算机节点的存储集群中,每个计算机节点包括 CPU 和至少为零的磁盘驱动器并执行其自己的操作系统,使所述计算机节点和所述磁盘驱动器闲置的方法,所述方法包括:

对于所述存储集群中的每个计算机节点,确定在第一预定量的时间内所述每个节点没有出现来自外部客户应用的客户请求;

对于所述存储集群中的每个计算机节点,确定在第二预定量的时间内所述每个节点没有出现来自所述存储集群中的其他节点的写入或读取请求;

对于所述存储集群中的每个计算机节点,当确定在所述第一预定量的时间和所述第二预定量的时间内既没有出现所述客户请求也没有出现写入请求时,独立地使所述每个计算机节点转变到闲置状态;以及

当确定每个磁盘卷的工作队列已经空了第三预定量的时间时,使闲置的计算机节点的所述每个磁盘卷转变到闲置状态。

2. 根据权利要求 1 所述的方法,其中,所述第一预定量的时间、所述第二预定量的时间和所述第三预定量的时间是相同的值。

3. 根据权利要求 1 所述的方法,其中,所述方法还包括:

使已经转变到闲置状态的每个磁盘卷停止旋转。

4. 根据权利要求 1 所述的方法,其中,所述存储集群是一次写入、多次读取 (WORM) 式固定内容存储集群。

5. 根据权利要求 1 所述的方法,其中,所述方法还包括:

通过停止在所述每个计算机节点的磁盘卷上执行的所有文件完整性检查来使所述每个计算机节点闲置;以及

使所述每个计算机节点的所述操作系统降低所述每个计算机节点的所述 CPU 的时钟速度;

6. 根据权利要求 1 所述的方法,还包括:

通过引导闲置的计算机节点的每个磁盘卷开始旋转减慢来使所述每个磁盘卷闲置。

7. 根据权利要求 1 所述的方法,还包括:

通过计算竞标值来使每个计算机节点闲置,其中所述竞标值减小所述每个计算机节点在与活动的计算机节点的竞争中赢得对于外部读取或写入的竞标的机会。

8. 一种使具有多个相互连接的计算机节点的存储集群内的计算机节点闲置的方法,所述方法包括:

通过所述计算机节点处理来自处于所述存储集群外部的客户应用的客户请求,所述计算机节点处于活动状态;

确定在第一预定量的时间内所述计算机节点没有出现来自外部客户应用的客户请求;

确定在第二预定量的时间内所述计算机节点没有出现来自所述存储集群中的其他计算机节点的读取或写入请求;

独立于所述存储集群中的所述其他计算机节点使所述计算机节点转变到闲置状态;以及

当确定每个磁盘卷的工作队列已经空了第三预定量的时间时,使闲置的计算机节点的

所述每个磁盘卷转变到闲置状态。

9. 根据权利要求 8 所述的方法,还包括:

当确定在所述第一预定量的时间内在所述每个其他计算机节点没有出现来自外部客户应用的客户请求并且在所述第二预定量的时间内在所述计算机节点没有出现来自所述存储集群中的其他计算机节点的读取或写入请求时,执行所述转变所述存储集群中的所述其他计算机节点中的每个的步骤。

10. 一种读取来自具有多个计算机节点的存储集群的计算机文件的方法,所述方法包括:

在所述存储集群的第一计算机节点接收来自外部客户应用的从所述存储集群检索所述计算机文件的请求,所述存储集群的所述计算机节点中的每个都处于闲置状态;

从所述第一计算机节点发送组播消息给所述存储集群中的所述计算机节点,所述组播消息请求对读取所述计算机文件的竞标;

基于对所述组播消息的响应选择所述多个计算机节点中的一个;

将所述选择的计算机节点和所述选择的计算机节点的包括所述计算机文件的磁盘卷转变到活动状态;

保持所述存储集群的未被选择的计算机节点处于所述闲置状态;以及

使来自所述选择的计算机节点的所述磁盘卷的所述计算机文件返回到所述外部客户应用。

11. 根据权利要求 10 所述的方法,其中,所述存储集群是一次写入、多次读取 (WORM) 式固定内容存储集群。

12. 根据权利要求 10 所述的方法,其中,所述闲置的计算机节点中的每个都具有 CPU 的降低的时钟速度。

13. 根据权利要求 10 所述的方法,其中,所述闲置的计算机节点的每个磁盘卷都处于旋转减慢的过程中。

14. 一种将计算机文件写入到具有多个计算机节点的存储集群中的方法,所述方法包括:

在所述存储集群的第一计算机节点接收来自外部客户应用的将所述计算机文件写入到所述存储集群中的请求,所述存储集群的所述计算机节点中的每个都处于闲置状态;

从所述第一计算机节点发送组播消息给所述存储集群中的所述计算机节点,所述组播消息请求对写入所述计算机文件的竞标;

基于对所述组播消息的响应选择所述多个计算机节点中的一个;

将所述选择的计算机节点和所述选择的计算机节点的磁盘卷转变到活动状态;

保持所述存储集群的未被选择的计算机节点处于所述闲置状态;以及

将来自所述外部客户应用的所述计算机文件写入到所述选择的计算机节点的所述磁盘卷中。

15. 根据权利要求 14 所述的方法,其中,所述存储集群是一次写入、多次读取 (WORM) 式固定内容存储集群。

16. 根据权利要求 14 所述的方法,其中,所述闲置的计算机节点中的每个都具有 CPU 的降低的时钟速度。

17. 根据权利要求 14 所述的方法,其中,所述闲置的计算机节点的每个磁盘卷都处于旋转减慢的过程中。

18. 一种将计算机文件写入到具有多个计算机节点的存储集群中的方法,所述方法包括 :

在所述存储集群的第一计算机节点接收来自外部存档应用的将所述计算机文件写入到所述存储集群中的请求 ;

从所述第一计算机节点发送组播消息给所述存储集群中的所述计算机节点,所述组播消息请求对写入所述计算机文件的竞标 ;

所述计算机节点中的每个计算用于获得写入所述计算机文件的机会的竞标值,来自具有较大容量的计算机节点的竞标值比来自具有较小容量的计算机节点的竞标值高 ;

选择所述多个计算机节点中具有最低竞标值的一个 ;以及

将来自所述外部客户应用的所述计算机文件写入到所述选择的计算机节点的磁盘卷中。

19. 根据权利要求 18 所述的方法,其中,所述存储集群是一次写入、多次读取 (WORM) 式固定内容存储集群。

20. 根据权利要求 18 所述的方法,还包括 :

将所述计算机文件写入到所述选择的计算机节点的具有最低容量的磁盘卷。

21. 根据权利要求 18 所述的方法,其中,所述存储集群的所述计算机节点中的每个都处于存档模式,其中当所述计算机节点中的每个都不处于存档模式时,计算出的来自具有较大容量的计算机节点的竞标值小于来自具有较小容量的计算机节点的竞标值。

22. 一种将计算机文件存储在具有多个计算机节点的存储集群中的方法,所述方法包括 :

接收来自所述存储集群中的第一计算机节点的组播消息,所述组播消息请求对写入所述计算机文件的竞标,所述计算机文件源自外部存档应用 ;

判断所述存储集群中的第二计算机节点是否处于存档模式 ;

当确定所述第二计算机节点处于存档模式时,计算用于获得写入所述计算机文件的机会的竞标值,其中所述计算机节点的容量越大,所述竞标值则越高 ;

基于所述计算的竞标值在所述计算机节点中赢得竞标 ;以及

将来自所述外部客户应用的所述计算机文件写入到所述第二计算机节点的磁盘卷中。

23. 根据权利要求 22 所述的方法,其中,所述存储集群是一次写入、多次读取 (WORM) 式固定内容存储集群。

24. 根据权利要求 22 所述的方法,还包括 :

将所述计算机文件写入到所述选择的计算机节点的具有最低容量的磁盘卷。

25. 根据权利要求 22 所述的方法,还包括 :

将所述第二计算机节点的磁盘卷填充到接近容量 ;以及

当确定预定量的时间已经过去并且没有处理过来自所述外部存档应用的请求时,使所述第二计算机节点闲置。

26. 一种在具有多个计算机节点的存储集群中处理外部请求的方法,所述方法包括 :

在所述存储集群的闲置计算机节点接收组播消息,所述组播消息请求对处理所述外部

请求的竞标,所述外部请求源自外部客户应用;

计算处理所述外部请求的机会的竞标值;

判断所述存储集群中的所述闲置计算机节点是否处于闲置状态,其中当确定所述闲置计算机节点处于闲置状态时,向所述竞标值添加补贴值,从而产生最终竞标值;

提交所述最终竞标值;

接收由于所述最终竞标值高于另一竞标值而使所述最终竞标值未被接受处理所述外部请求的通知;以及

不由所述闲置计算机节点处理所述外部请求。

27. 根据权利要求 26 所述的方法,其中,所述存储集群是一次写入、多次读取 (WORM) 式固定内容存储集群。

28. 根据权利要求 26 所述的方法,还包括:

通过对所述提交的最终竞标值没有接收到响应来接收通知。

29. 根据权利要求 26 所述的方法,还包括:

保持所述闲置计算机节点处于闲置状态,其中所述闲置状态的特征在于所述闲置计算机节点的 CPU 的时钟速度降低。

30. 根据权利要求 1 所述的方法,还包括:

对所述节点中包括硬盘驱动器的第一节点计算竞标值,该竞标值减小所述节点中的所述第一节点在与活动的计算机节点的竞争中赢得对于外部读取或写入的竞标的机会;以及

对所述节点中包括固态驱动器的第二节点计算竞标值,该竞标值增大所述节点中的所述第二节点在与活动的计算机节点的竞争中赢得对于外部读取或写入的竞标的机会。

31. 根据权利要求 8 所述的方法,还包括:

对于包括磁盘驱动器的每个磁盘卷,调节写入或读取竞标值,以减小包括所述磁盘驱动器的所述卷在与活动的计算机节点的竞争中赢得对于外部写入或读取的竞标的机会;以及

对于包括固态驱动器的每个磁盘卷,调节写入或读取竞标值,以增大包括所述固态驱动器的所述卷在与活动的计算机节点的竞争中赢得对于外部写入或读取的竞标的机会。

32. 一种将计算机文件写入到具有多个计算机节点的闲置存储集群的方法,所述存储集群包括多个驱动器、至少一个固态驱动器以及至少一个硬盘驱动器,所述方法包括:

在所述存储集群的第一计算机节点接收来自外部应用的将所述计算机文件写入到所述存储集群中的请求,所述闲置存储集群的所述多个驱动器全部都处于闲置状态;

从所述第一计算机节点发送组播消息给所述存储集群中的所述计算机节点,所述组播消息请求对写入所述计算机文件的竞标;

所述计算机节点中的每个都计算写入所述计算机文件的机会的竞标值,其中来自包括所述至少一个固态驱动器的第一计算机节点的第一竞标值低于来自包括所述至少一个硬盘驱动器的任何其他计算机节点的第二竞标值;

选择具有最低竞标值的所述第一计算机节点;以及

将所述计算机文件从所述外部客户应用写入到所述选择的第一计算机节点的所述至少一个固态驱动器上。

33. 根据权利要求 32 所述的方法,其中,所述存储集群是一次写入、多次读取 (WORM) 式

固定内容存储集群。

存储集群中的自适应功率保存

技术领域

[0001] 本发明总的涉及功率保存。更具体地，本发明涉及存储集群中的功率保存。

背景技术

[0002] 当前，正在进行普遍的努力以减小计算机的电力消耗，从而推动所谓的“绿色”技术并解决全球气候变化。但是，除了这些长期目标之外，减小的功率消耗还能够为企业主提供立即见效的节省。计算机系统使用的功率的减少不仅意味着计算机系统自身的电费降低，而且功率减小还将意味着计算机系统的冷却成本减少。对于管理大量数据中心的企业，冷却计算机机架所需的功率的减小会具有非常重要意义。对于这些企业，冷却容量(capacity)的减小还意味着数据中心所需的冷却基础设施设置减少以及所需的不动产减少。

[0003] 因此，通常理想的是保存计算机存储集群中的功率，以节省钱并减少必需的冷却。但是，在客户应用一直向集群写入并从集群读取的存储集群中并且在存储集群自身评价其自己的数字对象完整性并进行数字对象的必要拷贝时，设计功率保存方案是具有挑战性的。

[0004] 因此，需要这样一种技术和系统：其将减少计算机集群中的功率消耗，同时仍允许客户应用进行必要的访问，并将维持集群内的对象完整性。

发明内容

[0005] 为了实现以上所述，并且根据本发明的目的，公开了一种功率保存技术。可以理解的是通过在卷(volume)不在使用中时使磁盘驱动器旋转减慢以及通过在节点不在使用中时减少CPU使用来保存功率，对存储集群是有利的。

[0006] 本发明提出了一种机制，通过该机制，使用不足的集群能够开始使磁盘旋转减慢，并减少节点CPU的使用。目标不是特意地减少可用的处理能力以保存功率，而是辨识处理能力要求何时较低并采取行动以保存功率。在夜间或周末休眠的集群或者对于写入/读取/删除操作不常使用的集群将在休眠时期期间享受到显著的功率使用减小，或许高达80%至90%。集群的后续访问将自动地唤醒一个或多个节点来执行请求并最终可以恢复整个集群。尽管在对睡眠节点的首次访问时将出现附加的等待时间，但可得到的最大处理能力将不受影响。

[0007] 该方法的优点包括：本发明能够根据客户应用的大幅变化的使用方式进行调节；每个节点和卷关于何时睡眠和苏醒作出其自己各自的决定；以及整个集群可以完全休眠。本发明不需要管理员设计更复杂的子集群布置以适应功率保存；并且，存在基本为零的管理日常支出。

[0008] 本发明的第一实施例允许存储集群中的计算机节点及其硬盘各自独立地闲置，从而保存功率。第二实施例允许在整个存储集群闲置时对存储集群中的相关节点的读取或写入访问而不需要唤醒整个集群。第三实施例使用用于存档应用的存档模式，在该模式下，某些节点保持闲置和空载，直至被需要。第四实施例允许在集群已经闲置预定时间之后发生

整个集群的健康处理。第五实施例使用固态驱动器 (SSD) 来消除由于旋转加快延时造成的闲置集群的初始等待时间。

附图说明

[0009] 通过参照结合附图进行的下面描述,本发明连同其进一步的优点可以得到最佳的理解,在附图中:

- [0010] 图 1 示出了用于本发明的操作的环境。
- [0011] 图 2 是流程图,描绘了在正常操作中客户应用如何将文件写入到存储集群中。
- [0012] 图 3 是流程图,描绘了节点如何从活动状态转变到闲置状态。
- [0013] 图 4 是流程图,描绘了闲置的集群如何处理读取请求。
- [0014] 图 5 是流程图,描绘了闲置的集群如何处理写入请求。
- [0015] 图 6A 和 6B 示出了适于执行本发明的实施例的计算机系统。

具体实施方式

[0016] 本发明可以利用任何合适的计算机硬件和软件来执行。存储集群可以包括任何数量的计算机节点。优选地,每个节点包括 CPU(或多个 CPU)、操作系统、通向其他节点(或至少中央路由器)的通信链路、以及任何数量的内部硬盘驱动器或固态驱动器(即,从零到 N 个)。一般地,每个节点包括至少一个驱动器,并且可以存在硬盘驱动器与固态驱动器的任何组合。集群一般为固定内容存储集群,这意味着其用于备份、长期存储、存档等,并且一般不用于对计算机文件的每日访问。通常称为 WORM(一次写入,多次读取)存储器,这意味着一旦计算机文件或数字对象被写入到集群中,便不能被改变。(当然,计算机文件的修改版本也可以存储在集群内)。集群可以执行为独立节点冗余阵列(RAIN),这意味着每个节点运行其自己的操作系统并作出关于集群内的存储的独立的决定。存储集群可以构建在片式机、塔式机、个人计算机和服务器上。可替代地,单个计算机箱内的多核处理器可以支持在每个核上运行的虚拟存储节点,这意味着具有多个节点的存储集群可以存在于单个计算机箱内。另外,单个物理箱内的计算机系统可以包括多个 CPU,在这种情况下,每个 CPU 可以表示节点,并且存储集群可以在该单个物理箱内执行。

[0017] 图 1 示出了用于本发明的操作的环境 10。包括有存储集群 20、客户应用 30、管理控制台 40、任何数量的计算机节点 51-54、以及中央路由器 70。如上所述,计算机节点是物理文件服务器,其优选地包括至少一个 CPU 和任何数量的磁盘驱动器、固态驱动器或具有这两种类型的混合驱动器。每个节点执行诸如 Debian Linux 的操作系统,并执行处理以管理节点之间的对等通信,从而执行健康的处理,并代表节点及其卷作出独立的决定。每个节点还包括管理软件,并且其状态能够通过网页浏览器在互联网上进行观察。

[0018] 在一个特别的 RAIN 实施例中,每个节点是 1U 服务器(例如,x86 服务器),具有带标准以太网联网能力的 1 太拉字节(terabyte) 或更多字节的串口 ATA 磁盘存储容量。每个节点具有 IP 地址并可以利用基于 IP 的 LAN、MAN 或 WAN 物理地相互连接。因此,每个节点可以与单个节点进行对话并可以利用路由器 70 或其他类似的网络开关将信息广播至存储集群内的所有节点(组播)。

[0019] 每个节点包括管理模块,该管理模块用于处理来自客户应用的外部请求(例如,

SCSP 请求)、节点间的复制请求(例如, SCSP 间请求)、以及其他节点间协议通信(竞标(bidding)、用于信息的请求,等等)。健康处理模块(将在下面描述)管理每个节点的数字内容。管理控制台 40 优选是链接到存储集群的网页服务器,其允许通过任何合适的互联网连接访问每个节点。每个节点执行能够用来观察和管理整个集群的冗余管理控制台。

[0020] 在一个实施例中,利用可以从德州奥斯丁的 Caringo 公司得到的 CASTor 内容存储软件或任何合适的计算机硬件来执行存储集群。在该实施例中,存储集群是固定内容存储器并且每个数字对象在集群内通过随机的数字(通用唯一标识符,或 UUID)被唯一地寻址,该 UUID 是利用真随机数字生成器对该数字对象生成的。利用散列函数对每个数字对象的内容进行验证。客户软件应用在将数字对象存储在集群中时接收 UUID 并通过将该 UUID 提供至集群来检索该数字对象。软件应用利用标准 HTTP 1.1,更具体地,利用称为简单内容存储协议(SCSP)的标准的简化子集与 CASTor 集群进行通信。利用该标准接口,客户应用(例如电子邮件、企业内容管理、健康护理应用、网页浏览器、Web 2.0、图片共享、社交媒体网站、安全视频、视频编辑,等等)可以访问 CASTor 存储集群。另外,直接 HTTP 访问对浏览器、JAVA、Python、C++ 和其他软件环境是可用的。

[0021] 综述

[0022] 本发明允许集群中的每个节点和卷基于该集群在可配置时期内实际处理来自外部客户应用的请求以及节点之间的复制请求的频率关于是否减少其功耗作出独立的决定。使用称为“休眠后(sleepAfter)”和“醒来后(wakeAfter)”的两个节点配置参数,这两个参数例如都是整数秒(这些参数不需要对于集群中的所有节点都具有相同的值)。

[0023] 如果节点没有在最后的 sleepAfter 秒内处理过来自客户应用的请求(读取或写入)或来自另一个节点的类似请求,其将开始休息。如果继续没有要求读取或写入的外部要求(例如,如在夜间或周末所预见的那样),那么最终节点及所有其卷将变为完全休眠。当客户应用一旦重新开始发送请求时,集群中的节点中的一个或多个将醒来,以处理这些请求。即使长时间没有检测到外部活动,每个节点也将在已经闲置了 wakeAfter 秒之后醒来,使其能够周期性地执行其健康处理事务(确保数字对象的至少最小数量的指定拷贝存在于集群内)。

[0024] 使用竞标处理,每个节点通过该竞标处理计算竞标值,该竞标值反映了该节点写入数字对象、读取数字对象或复制数字对象(在该节点上保留拷贝的成本)的成本是多少。通常,节点通过向请求来自所有可用节点的竞标的初次访问节点投标而对这些写入、读取和复制请求进行竞标。竞标是数值,并且竞标越低,节点赢得处理请求的机会的机会则越大。写入是从客户应用取得数字对象并将其存储在集群的节点中的磁盘上的过程。读取是接受来自客户应用的用于数字对象的唯一标识符、在集群中的磁盘上找到数字对象、以及然后将该对象返回到客户应用的过程。复制请求确定将数字对象的拷贝存储在集群内的节点上的成本(假定复制因子等于 2,那么集群中需要两个拷贝的数字对象)。复制请求之后的竞标结果可以涉及拷贝和存储对象的另一个拷贝。为了保存集群中的功率,竞标将考虑节点是否闲置、节点的容量、节点在过去几分钟内有多忙、有哪些类型的磁盘驱动器可用、以及节点的当前状态。

[0025] 具有最低竞标的节点赢得写入、读取或复制的机会。复制竞标反映了将数字对象的一个拷贝保持在其所在处的成本。健康处理将使用复制竞标来决定是否移动一个拷贝到

不同的节点。基本上,健康处理向每个节点询问 :是否具有该数字对象的一个拷贝以及保持其在该处的成本是多少。每个节点通过计算其复制竞标并将其提供给健康处理模块来进行响应。

[0026] 闲置节点通过暂停其健康处理并改变其竞标 (将在下面描述) 以使其优选不读取、写入或复制数据流给另一个节点来减少其功耗。其操作系统将注意到使用减少并将相应地降低 CPU 时钟速度和功耗。以高时钟速度运行处理器允许更好的性能。然而,当同一处理器在较低频率 (速度) 下运行时,其产生较少的热并消耗较少的功率。在很多情况下,还能够降低核心电压,从而进一步减少功耗和热生成。这能够保存存储服务器中的电池功率,延长处理器寿命,并且减少可变速风扇产生的噪声。大多数英特尔芯片组都具有 SpeedStep ® 技术以在处理器要做的工作较少时自动地降低 CPU 时钟速度。来自 AMD 的称为 PowerNow ! ® 的类似技术完成对该公司的芯片组同样的用途。

[0027] 闲置的卷主要通过断开磁盘 I/O 以及使其磁盘旋转减慢来降低其功耗。当节点变为闲置时,一些或全部其卷可能仍然是活动的,并且工作于未完成的节点任务。卷仅当其在过去的 sleepAfter 秒内已经没有 I/O 活动时才转变到闲置状态并使其磁盘旋转减慢。

[0028] 详细设计

[0029] 使用四个集群范围配置参数 :sleepAfter, 即在活动节点或卷变为闲置之前不活动的秒数 (缺省为 2 小时) ;wakeAfter, 即在闲置节点重新变为活动之前不活动的秒数 (缺省为 8 小时) ;idleCost (闲置成本), 即所增加的读取和写入用于闲置节点的竞标的竞标补贴 (bid premium) (缺省为 10) ;以及 replication factor (复制因子), 即存储集群内必须保持的每个数字对象的拷贝数 (缺省为 2, 但每个对象可以含有构建该参数的元数据)。这些参数存储于在启动时间发送至每个节点的集群配置文件中。

[0030] 存档模式 (archive mode) 设定被提供给每个节点,用于执行存档的那些应用。该设定 (一般为二元值) 通过从 100 减去写入竞标值以获得新的写入竞标值来改变节点上的所有卷的正常写入竞标值。在使用时,该设定限定节点如何竞标写入,试图尽可能长时间地保持节点休眠。真正的存档应用对写入和读取处理能力没有特别高的要求,特别是与非常大的集群的总处理能力容量相比。用于计算写入竞标和复制竞标的正常竞标算法试图在集群中的所有节点之间统一地加载平衡以使处理能力最大化,但是该平衡具有不期望的剥夺节点的休眠的副作用。集群管理员可以选择指定一些或所有节点处于存档模式。一般地,足够大以处理峰值处理能力要求的节点的核心组将是存档的。完全为了增加容量而增加的新的节点能够被指定为存档节点并将一直休眠,直至被需要。

[0031] 对于这些应用,存档模式设定对于写入请求基本上改变节点上的所有卷的正常竞标样式。空载或几乎空载的卷将优选不写入数据流,而较满的卷极度地试图存储越来越多的数据,直到其临达到容量之前。节点在要求其他较空的卷变为活动之前将试图填满其几乎满载的卷。结果是 :新的相对较空的节点将不能赢得任何写入竞标,将不处理客户应用写入请求,并将继续休眠,这是存档应用中理想的。在本公开中描述的功率保持机制的情况下并假定这种类型的存档改变竞标,期望添加到集群的新的节点或卷保持在低功率模式下闲置,直至实际需要其容量。较早的节点和卷将继续被使用,直至其变的几乎满载,此时,这些节点和卷也将开始休眠。

[0032] 为了保存功率,每个节点都能够转变到闲置状态。如果节点在最后的 sleepAfter

秒内没有检测到来自其他节点的进行读取或写入的客户应用请求或类似的请求,那么其将进入闲置状态;在该状态下,节点开始减少其CPU使用并改变其竞标措施以保存功率。前者简单地通过在处于闲置模式时使节点的健康处理模块暂停来实现。CPU使用的减少将通过自动地降低CPU和RAM的时钟速度使操作系统和CPU自身保存功率,使用较少的功率,产生较少的热,并且使冷却风扇也能够较慢地运行并使用较少的功率。

[0033] 如上所述,节点使用竞标来与其他节点竞争。将在下面描述的竞标变化旨在使闲置节点对于读取、写入、删除和需要将文件从闲置节点移动的复制(即,需要磁盘访问的动作)赢得将来的竞标的可能性较低。如果集群中的其他节点仍然是醒着的,那么这些节点将更可能处理这些请求,从而允许闲置节点保持休眠。注意,闲置节点仍响应于节点间协议消息,并还可继续工作于未完成的工作,如已经排队等候的复制请求和来自其自己的健康处理器的请求以读取和检查对象的完整性。

[0034] 闲置节点在其健康处理暂停并且其竞标是“闲散的”这一意义上来说是闲置的。其将在被要求这样做的情况下继续用作主访问节点(PAN, Primary Access Node),并将继续响应于来自其他节点的节点间协议询问,所有这些询问都仅仅利用存储器内的结构(而不是访问磁盘)来回答。作为PAN来处理外部请求将不唤醒休眠的节点(转变到活动状态),除非该节点实际上赢得了竞标。当节点处于闲置状态时,其将如下地限定对于写入、读取(包括删除)和复制的其正常节点间竞标:写入竞标—照常计算,然后添加idleCost(竞标成本);读取竞标—照常计算,然后添加idleCost;复制竞标—照常计算,然后减去idleCost。即,将已有的副本保持在闲置节点上的适当位置比将其从闲置节点移走的成本要低。

[0035] 总之,读取闲置节点或写入到闲置节点的成本较高,而保持已有的副本在其位置处成本较低。注意,健康处理使用写入竞标来决定是否将副本移动到闲置节点上,并使用复制竞标来决定是否从闲置节点移走副本。限定竞标(而非仅仅不对读取和写入竞标)的原因是为了避免由在其他节点保持醒着并处理请求时休眠的一些节点所导致的阻塞点。尽管这可能在一定时期内发生,但最终,由醒着的节点返回的竞标将超过来自休眠节点的竞标,甚至还考虑了闲置补贴的因素。

[0036] 闲置节点将仍参与节点间协议通信,并可仍具有忙碌的卷。在不存在外部客户应用请求、内部复制活动和健康处理循环的情况下,闲置节点将继续处理某些来自其他节点的请求。如果所述其他节点中的一些仍然活动,则存在大量需要访问磁盘卷以进行响应的节点间消息。这些节点间消息包括:缩减请求—其在存在太多副本时由另一个节点发送;删除询问—其在另一个节点响应于动态删除而删除了数据流之后被发送;版本询问—其由另一个节点发送,以询问固定流(anchor stream)的版本数量;以及版本通知—其在确定了存在陈旧的易变对象时由另一个节点发送。由于这些节点间请求将需要被处理,所以闲置节点一般将不能使其所有卷都立即旋转减慢。最终,集群中的大多数或全部的其他节点也将变为闲置,并且节点间活动将减少,并最终全部停止。当这发生时,闲置节点的卷将开始自己变为闲置。

[0037] 节点的每个卷也具有闲置状态。单个卷将独立地决定何时其应当使其磁盘旋转减慢。卷使用与节点整体使用的相同的标准和参数来这样做。一旦卷注意到其有序的队列上已经sleepAfter秒没有磁盘活动了,其将转变到卷闲置状态并使其磁盘旋转减慢。与节点

不同,卷不需要每个 wakeAfter 秒都醒着,因为节点本身将醒来并且开始在其卷上重复循环,从而自动地使磁盘重新再旋转加快。固态驱动器不物理地旋转减慢或旋转加快,而是可以仍然接受和处理作为转变到闲置状态或从闲置状态转变的一部分的“旋转减慢”和“旋转加快”命令(例如,驱动器可以仅仅确认该命令)。

[0038] 连接于其中一个节点的管理控制台显示集群状态。管理控制台还显示闲置节点和闲置卷状态。与基本从其卷的组合状态得到的其他节点状态不同,闲置状态对节点和卷是独立的。换句话说,节点可能在所有其卷都活动时处于闲置。还可能的是,活动节点在一段较短的时期具有闲置的卷。除了反映闲置状态之外,管理控制台还支持在运行时间从集群状态页面上的设定窗口修改对于 sleepAfter 和 wakeAfter 参数的集群范围设定。由完全闲置的存储集群处理的第一读取或写入请求可能在一个或多个卷醒来并使其硬盘旋转加快时经历异常长的等待时间。后续的操作将很可能返回到此时旋转的卷,如上所述,因此该附加的等待时间仅仅出现在第一次请求时。

[0039] 固态驱动器(SSD)能够被用来减小或消除该附加的等待时间,因为当闲置与活动基本相同时,与旋转介质及其等待时间相比,固态驱动器已经是功率高效的,即,固态驱动器并不物理地旋转减慢。然而,与常规的磁盘驱动器相比,SSD 一般成本更高,并且具有较小的容量。为了对这些不同的特性进行建模,SSD 可以与普通的驱动器不同地计算其竞标。SSD 通常将具有较低的写入竞标和读取竞标,反映了其较高的处理速度,并具有较高的复制竞标,反映了其较小的容量和较高的成本。换句话说,SSD 将优选初始存储新的数据,但是尽快将其复制给其他成本较低的介质。来自 SSD 的竞标(与来自节点的其他驱动器的竞标一起)将在计算用于整个节点的竞标值时被其节点考虑在内。

[0040] 重要地,SSD 将不像常规的卷那样在处于闲置模式时调节其竞标。由于该行为,闲置集群中的 SSD 卷将最有可能对其存储的数据赢得任何写入或读取竞标。总的效果在于,含有 SSD 的闲置集群将以非常短的等待时间——与没有 SSD 的闲置集群相比——响应于请求,而不会负面影响功率节省特性。

[0041] 健康处理

[0042] 如上所述,每个节点的健康处理模块在节点的数字对象上重复循环,以确保所有数字内容的完整性,从而确保每个数字对象的适当数量的拷贝存在于存储集群内,并将对象移至成本较低的节点以保留处理能力。当数字对象被客户应用首次写入到存储集群时,用于该对象的散列值被计算并被存储在可由管理存储集群的软件访问(例如,可由健康处理模块访问)的存储位置中。可以使用任何合适的散列函数,例如,MD5 算法非常适于该目的。注意,该唯一的散列值用来确认每个数字对象的完整性,并不一定与用来定位数字对象的通用唯一标识符相同。在优选实施例中,用于完整性目的的 MD5 不同于通用唯一标识符。集群中每个节点的缺省状态是执行完整性检查,假定没有写入或读取(“送”或“取”)正在进行。每个节点的健康处理模块然后排查节点的所有数字内容,验证每个数字对象的原始存储的 MD5 等于对该数字对象新计算的 MD5。

[0043] 为了确保每个数字对象的适当数量的拷贝存在于存储集群内,健康处理模块使用组播技术。在一个实施例中,在完整性检查期间无论何时文件被接触到,健康处理模块都向所有的其他节点发送消息,询问其是否具有该数字对象以及其继续存储该数字对象的成本是多少。对该组播消息的响应将指示存储集群内是否需要更多拷贝的特定文件,并还将指

示在每个节点上继续存储这样的拷贝的成本（复制竞标）。

[0044] 即使存储集群内存在数字对象的足够拷贝（如数字对象本身任何元数据所限定的，由集群参数“复制因子”确定），健康处理模块可决定将对象移动到成本较低的节点。尽管该移动可以帮助确保高处理能力（数据传播开并且更多的节点活动），但这并不一定对存档应用是最佳的。存档应用在仅仅几个节点活动并存储数据的情况下更好地保存功率。一旦这几个节点满了，则其能够变休眠，从而节省功率。在一定情形下可以使用上面的存档模式参数。

[0045] 详细流程图

[0046] 图 2 是流程图，描绘了在正常操作下客户应用如何将文件写入到存储集群中。在步骤 204 中，任何合适的客户应用 30 都期望将数字对象（例如任何计算机文件、数字图像、数字电影、健康记录等）存储在存储集群 20 中。客户应用发现或获得集群内的节点 51 中的一个的 IP 地址并选择该节点作为主访问节点 (PAN)，利用该主访问节点来开始存储过程。一般地，客户应用将访问其下一个写入请求所使用的最后节点。在步骤 208 中，客户应用发送请求 201 给 PAN 以存储数字对象。在一个实施例中，该请求是 HTTP POST 请求，该请求包括数字对象的字节的标头和长度。对从 PAN 到客户的该请求的响应是以下几种中的一种：“是”，PAN 能够存储对象；“否”，此处为存储对象的更好的节点；或者“否”，该集群内没有节点能够存储该对象。

[0047] 在步骤 212 中，PAN 接收该请求并通过向集群内的所有节点发送组播消息进行响应，其中该组播消息指示要存储的数字对象及其字节的大小，以确定将使用哪个节点来存储对象。在步骤 216 中，每个节点（包括 PAN）计算对于该数字对象的写入竞标并将该竞标值返回给 PAN。写入竞标值反映了将数字对象存储在每个节点上的成本并在当前被执行行为从 0 到 100 的值。较低的竞标意味着将对象存储在该节点上成本较低并且该节点更有可能赢得竞标。用来计算用于节点的写入竞标值的因素包括：节点有多满，数字对象的大小，节点是活动的还是闲置的，节点近期有多忙碌，哪些类型的卷是可用的（固态驱动器，磁盘驱动器，等等）及其各个竞标，以及卷的状态（休息，闲置，等等）。换句话说，节点的竞标从其卷中的每个的各个竞标得到。节点对每个卷计算存储成本，采用最低（最佳）值，并且然后使用该最低值来如上所述地计算其自己的节点 - 水平竞标。节点可以具有很多具有不同竞标的驱动器，因为这些驱动器具有不同的容量，或者这些驱动器处于不同的状态，等等。

[0048] PAN 然后选择具有最低竞标值的节点，并且控制移动到步骤 220 或步骤 232。在步骤 220 中，PAN 具有最低的竞标并且其通过将“继续”消息发送回至客户应用来进行响应。作为响应，客户在步骤 224 中发送数字对象给 PAN，并且在步骤 228 中，PAN 存储数字对象，计算通用唯一标识符并将该标识符返回至客户应用。

[0049] 另一方面，如果 PAN 失去竞标，那么在步骤 232 中，PAN 将客户应用重新引导至然后将处理请求的具有最低竞标的节点 53(202)，即辅助访问节点 (SAN, Secondary Access Node)。然后在步骤 236 中，客户应用发送相同的请求给 SAN（如在步骤 208 中那样），并且节点在步骤 240 中通过将“继续”消息发送回至客户应用来进行响应。作为响应，客户应用在步骤 244 中发送数字对象给 SAN，并且在步骤 248 中，SAN 存储数字对象，计算通用唯一标识符并将该标识符返回至客户应用。

[0050] 以类似的方式，客户应用可以请求存储集群返回 UUID 所标识的特定数字对象。这

可以利用 SCSP GET 请求来实现。客户应用将标识符提供至存储集群的 PAN, 组播消息连同标识符被一起发送至集群内的所有节点, 每个节点计算读取竞标 (读取数字对象并将其返回至客户应用的成本), PAN 选择具有最低读取竞标的节点, 并且该节点然后将数字对象返回至客户应用。当然, 不存储数字对象的节点 (由节点的存储器内对象的唯一标识符的存在确定) 将不竞标。尽管任何节点都可以执行 POST 请求并向其驱动器中的一个写入数字对象, 但仅仅实际存储数字对象的节点才能够响应于来自客户应用的 GET 请求。节点还可以通过与以上所述相似的方式执行节点间写入和读取请求 (为了健康处理期间的复制目的)。

[0051] 因此, 实现了某些活动使节点“忙碌”并将其置于活动状态。如果节点响应于诸如写入文件或检索文件的外部客户请求、或者如果节点响应于另一个节点的写入或读取文件的请求 (或许作为健康处理的一部分), 那么节点是忙碌的 (并因此不处于闲置状态)。为了保存功率, 下面的技术使节点从活动状态转变到闲置状态。

[0052] 图 3 是流程图, 描绘了节点如何从活动状态转变到闲置状态。该流程由存储集群中的每个节点独立地执行。最初, 假定节点是忙碌的或活动的, 意味着其正在执行写入、读取和健康处理活动 (检查存储在节点处的每个文件的完整性, 判断集群内是否存在适当数量的文件拷贝, 并且将文件移至成本较低的节点)。当然, 如果发现文件损坏, 则应当被删除, 并且在集群内存储附加的拷贝。类似地, 如果健康处理判定在集群内没有发现适当数量的拷贝, 那么也需要存储附加的拷贝。如果能够发现成本较低的节点, 健康处理还可将文件写入到另一节点 (即使存在足够的拷贝)。所有这些活动都是 CPU 密集的并会涉及磁盘访问。然后, 活动的节点响应于来自客户应用或来自其他节点的读取和写入请求。

[0053] 因此, 在步骤 304 中, 节点由于这些活动中的任何一个而在当前是活动的。步骤 308 和 312 优选并行地执行并测试在预定时长内是否存在过任何写入或读取活动或涉及写入到磁盘的任何健康处理。关于健康处理活动, 应当注意, 一些活动可能不涉及将文件写入磁盘。例如, 如果完整性检查未发现任何损坏对象, 那么不需要写入附加的拷贝。类似地, 如果在集群内存在适当数量的拷贝, 那么不需要写入附加的拷贝。最后, 如果健康处理未发现成本较低的节点, 那么不需要移动文件的拷贝。涉及写入到磁盘的健康处理活动将阻止节点向闲置状态转变, 而不涉及写入的活动将不阻止该转变。

[0054] 如上所述, 参数 sleepAfter 可以设定为期望的时长。如果存在过任何活动, 那么节点保持活动, 但是, 如果在预定时长之后仍没有读取或写入活动或者没有任何涉及写入的健康处理活动, 那么在步骤 316 中, 节点转变到闲置状态。

[0055] 可以通过不同的方式来执行节点向闲置状态的转变。优选地, 转变涉及停止所有的健康处理活动, 从而保留非常少的活动处理。一旦操作系统注意到没有 CPU 活动 (或有非常少的 CPU 活动), 操作系统将降低 CPU 和 RAM 上的时钟速度。存储集群内的功率保存的一个优点在于, 除了外部客户应用请求、健康处理活动以及集群中的组播消息的处理之外, 存在非常少的其他活动处理 (如果存在的话)。一旦节点转变到闲置状态并且所有健康处理停止, 更可能的是操作系统将由于非常少的处理保留而降低时钟速度。在正常的计算机上, 存在大量始终运行的处理, 这些处理减少了操作系统降低时钟速度的可能性。一旦处于闲置状态, 节点便以“闲散”方式竞标, 从而减少其被要求执行读取、写入或复制的机会。此时, 节点是闲置的, 但是其任何卷都可以仍然处理其工作队列中未完成的工作。每个卷保持

其自己的工作队列。

[0056] 一旦在步骤 320 中确定用于特定卷的工作队列已经空了 sleepAfter 秒（即，磁盘没有事情可做），那么在步骤 328 中，该卷也将转变到闲置状态。如果用于卷的工作队列不是空的，那么在步骤 324 中，卷仍然忙碌，但是节点将保持闲置。卷通过利用任何适当的命令使其磁盘旋转减慢来转变到闲置状态。例如，一种控制磁盘旋转减慢的方法是通过安装在嵌入式 Linux 操作系统中的硬盘显示参数 (hdparm) 实体进行的。关注的命令行选项是 S、y、C、Y 和 Z。选项“S”设定用于驱动器的待机（旋转减慢）超时。该值由驱动器用来确定在关闭心轴马达以节省功率之前需要等待多久（没有磁盘活动）。步骤 320 中的测试可以利用操作系统执行或在硬件中执行。最后，一旦磁盘已经旋转减慢，其便在步骤 332 中停止，从而进一步减少功耗。

[0057] 图 4 是流程图，描绘了闲置的集群如何处理读取请求。在步骤 404 中，假定整个存储集群都是闲置的，也就是说，集群的每个计算机都处于闲置状态。另外，所有的健康处理活动都已停止。

[0058] 在步骤 408 中，在主访问节点 (PAN) 接收来自客户应用的外部读取请求，该请求请求由唯一标识符所标识的特定数字对象。在步骤 412 中，PAN 开始组播竞标过程。每个合格的节点计算其自己的读取竞标并将该值返回至 PAN。利用每个节点的 CPU 和 RAM 进行竞标值的计算并且没有必要进行磁盘访问，因而保存功率。节点将仅在其卷中的一个实际存储所需的数字对象的情况下才进行竞标；节点作出唯一标识符是否存储在节点的存储器中的该判断。用来计算用于节点的读取竞标值的因素包括：节点是活动的还是闲置的；节点近期有多忙碌；哪些类型的卷存储对象及其各个竞标；以及卷的状态（休息，闲置，等等）。如上所述，如果节点是闲置的，闲置成本值则被添加到竞标中。该附加值帮助闲置的节点保持闲置。如果所有的节点都是闲置的，那么一个节点将需要被唤醒以执行请求。

[0059] 在步骤 416 中，PAN 基于最低竞标选择响应的节点中的一个；此时，被选择的节点（以及所有的其他节点）仍然处于闲置状态。在步骤 420 中，所选择的节点从闲置状态转变到活动状态，并且其存储数字对象的卷也转变到活动状态。对于节点和卷，转变到活动状态涉及首先使含有数字对象的卷旋转加快。用于新激活的节点的健康处理器将直到读取完成才开始。

[0060] 在步骤 424 中，活动节点然后执行 GET 请求并将所识别的数字对象返回到客户应用。此时，在步骤 428 中，所选择的节点和卷保持处于活动状态，并且健康处理可以重新发生。接下来，节点开始执行图 3 的流程图，由此，节点可以最终转变回到闲置状态。

[0061] 图 5 是流程图，描绘了闲置集群如何处理写入请求。在步骤 504 中，假定整个存储集群都是闲置的，也就是说，集群的每个计算机节点都处于闲置状态。另外，所有的健康处理活动都已停止。

[0062] 在步骤 508 中，在主访问节点 (PAN) 接收来自客户应用的外部读取请求，该请求请求将特定数字对象写入到存储集群中。在步骤 512 中，PAN 开始组播竞标过程。集群的每个节点计算其自己的写入竞标（例如，如上所述）并将该值返回至 PAN。利用每个节点的 CPU 和 RAM 进行竞标值的计算并且没有必要进行磁盘访问，因而保存功率。如上所述，如果节点是闲置的，闲置成本值则被添加到竞标中。该附加值帮助闲置的节点保持闲置。如上所述，获胜的竞标将主要由节点及其卷的使用确定。具有更多可用存储容量的卷更可能赢

得竞标,如具有更多不使用的存储器的节点。

[0063] 在步骤 516 中, PAN 基于最低写入竞标选择节点中的一个;此时,被选择的节点(以及所有的其他节点)仍然处于闲置状态。在步骤 520 中,所选择的节点从闲置状态转变到活动状态,并且其最不满的卷也转变到活动状态。所选择的卷将使其磁盘旋转加快。

[0064] 在步骤 524 中,活动节点然后执行 POST 请求并将数字对象写入到最不满的卷,计算用于该数字对象的唯一标识符,并将该标识符返回至客户应用。此时,在步骤 528 中,所选择的节点和卷保持处于活动状态,并且健康处理可以再次发生。由于在存储集群内一般需要数字对象的至少两个拷贝,所以在步骤 532 中,辅助访问节点 (SAN) 再次发送组播消息,以再次开始竞标过程,为了获得写入数字对象的拷贝的机会。在步骤 536 中,一旦节点(复制竞争对手)赢得竞标,节点及其最不满的卷便转变到活动状态。然后,在步骤 540 中执行对象的第二拷贝到该其他节点中的写入。当然,如果用于存储集群的复制因子具有大约 2 的值,那么此时将写入数字对象的更多拷贝。接下来,两个节点开始执行图 3 的流程图,由此,这些节点中的每个可以最终转变回到闲置状态。

[0065] 因此,图 4 和 5 示出了休眠的集群如何可以在无需唤醒集群内的每个节点的情况下执行来自客户应用的写入和读取请求。

[0066] 示例

[0067] 假定 sleepAfter 为 30 分钟并且 wakeAfter 为 8 小时。在周五下午的 5:30 分,每个人回家过周末并且另外忙碌的存储集群发现自身没有新的工作要做。所有的节点将继续重复其内容,执行健康处理检查并进行必要的复制,直至集群中的所有数据流都是健康的并被完全复制。当达到该状态时,所有的复制活动都已经停止,尽管节点可能仍然忙于重复循环和检查、使用 CPU 循环以及访问硬盘。在某一时间点,其中一个节点注意到在过去的 30 分钟其没有新的工作可做,并且该节点将转变到闲置状态,从而暂停其健康处理并以“闲散”的方式进行竞标(即,将竞标调整为不希望读取、写入或复制数字对象)。其将继续响应于来自其他节点的节点间协议请求(包括涉及拷贝数量的请求),但是这些请求完全在存储器内结构之外被处理并且不需要磁盘访问。然而,新的闲置节点不向其他节点发送关于拷贝数量的询问,这是因为其健康处理不再重复循环。节点的卷一般在节点变闲置后一定时间内保持活动,同时其清理其工作队列。最终,卷将注意到其工作队列已经空了 30 分钟,并且将变为闲置,从而使其磁盘旋转减慢。

[0068] 其他节点将在 30 分钟后开始意识到其没有新的工作可做,并且也将变为闲置。最终,集群中的所有节点将变为闲置并停止发送任何节点间协议消息。休眠的集群中仅有的活动为“街头公告员 (town crier)”消息的不频繁的发送和接收,这些消息有助于集群检测休眠节点的故障。

[0069] 现在假定低频维护应用需要每小时将一个新的数据流(数字对象)写入到集群中。由于集群中的所有节点都是闲置的并且以闲散的方式进行竞标,所以其中一个节点将赢得写入竞标并将被迫醒来以处理请求。注意,主访问节点 (PAN, 被希望写入到集群中或从集群进行读取的客户应用首先访问的节点) 将不需要醒来,因为其不需要访问其磁盘以找到适当的 SAN(辅助访问节点,在主访问节点不能处理的情况下将实际处理客户请求的节点) 来重新引导请求。原因在于,节点能够在闲置的同时执行 PAN 活动。假定复制因子=2(集群内的每个数字对象的期望拷贝数量),那么 SAN 还将需要唤醒第二节点以用作其复

制竞争对手。这两个节点都将在接下来的 30 分钟内保持醒着,在这 30 分钟后,它们将再次变为闲置。

[0070] 如果另一个写入请求(来自客户应用)在两个醒着的节点返回到其休眠状态之前到达,那么其中一个将几乎确定地赢得竞标,这是因为所有的其他节点都在休眠。因此,例如每十分钟写入一次的客户应用将发现自己每次与相同的两个节点在对话,而集群的剩余节点继续保存功率。如果应用试图读取不在这两个节点中的一个上的数字对象,那么集群将仍然找到其副本,唤醒具有该副本的节点,并且读取数据。如果应用开始更频繁地操作或者要求更多的处理能力,越来越多的节点将醒来以处理增加的要求。

[0071] 即使在周末没有进一步的应用活动,每个节点也将每 8 个小时醒来一次并启动健康处理以检查其数据流并执行任何需要的校正动作。这将允许节点检测可能在其休眠期间变得不能操作的卷。如果在接下来的 30 分钟内不需要校正动作,那么节点将返回到闲置状态。

[0072] 计算机系统实施例

[0073] 图 6A 和 6B 示出了适于执行本发明的实施例的计算机系统 900。图 6A 示出了计算机系统的一种可能的物理形式。当然,计算机系统可以具有很多物理形式,包括:集成电路,印刷电路板,小型手持设备(如移动电话或 PDA),个人计算机或超级计算机。计算机系统 900 包括监视器 902、显示器 904、机箱 906、磁盘驱动器 908、键盘 910 以及鼠标 912。磁盘 914 是用来将数据传输到计算机系统 900 以及从计算机系统 900 传输数据的计算机可读介质。

[0074] 图 6B 是用于计算机系统 900 的框图的示例。多种子系统附接于系统总线 920。一个或多个处理器 922(也称为中央处理单元或 CPU)联接于包括存储器 924 在内的存储设备。存储器 924 包括随机存取存储器(RAM)和只读存储器(ROM)。如本领域所公知的,ROM 用来单向地向 CPU 传输数据和指令,而 RAM 一般用来以双向的方式传输数据和指令。这两种类型的存储器都可以包括将在下面描述的任何适当的计算机可读介质。固定磁盘 926 也双向地联接于 CPU922;其提供附加的数据存储容量并还可包括将在下面描述的计算机可读介质中的任何一种。固定磁盘 926 可以用来存储程序、数据等,并且一般是比主存储器慢的辅助存储介质(如硬盘)。应当理解,保留在固定磁盘 926 内的信息在适当的情况下可以以标准的形式结合为存储器 924 中的虚拟存储器。可移动磁盘 914 可以采用将在下面描述的计算机可读介质中的任何一种的形式。

[0075] CPU 922 还联接于多种输入/输出设备,如显示器 904、键盘 910、鼠标 912 和扬声器 930。通常,输入/输出设备可以是下列设备中的任何一种,这些设备为:视频显示器,轨迹球,鼠标,键盘,麦克风,触敏显示器,转换器读卡器,磁带读取器或纸带读取器,图形输入板,触笔,声音识别器或字迹识别器,生物特征读取器,或者其他计算机。可选地,CPU 22 可以利用网络接口 940 联接于另一个计算机或电信网络。利用这种网络接口,能够想到的是,CPU 可以在执行上述方法步骤的过程中接收来自网络的信息,或者可以向网络输出信息。另外,本发明的方法实施例可以完全在 CPU 922 上执行,或者可以结合共享处理的一部分的远程 CPU 在诸如互联网的网络上执行。

[0076] 此外,本发明的实施例还涉及具有计算机可读介质的计算机存储产品,所述计算机可读介质上具有用于执行各种计算机执行操作的计算机代码。所述介质和计算机代码可

以是为了本发明的目的而特别设计和构建的，或者其可以是计算机软件领域的技术人员所公知和可以得到的类型。计算机可读介质的示例包括但不限于：磁性介质，如硬盘、软盘和磁带；光学介质，如 CD-ROM 和全息摄像设备；磁 - 光介质，如软光盘；以及硬件设备，其特别构建成存储和执行程序代码，如专用集成电路 (ASIC)、可编程逻辑设备 (PLD) 以及 ROM 和 RAM 设备。计算机代码的示例包括：机器代码，如由编译器产生的机器代码；和含有较高水平代码的文件，所述较高水平代码由计算机利用解译器来执行。

[0077] 尽管为了理解清楚的目的在一定的细节上对上面的本发明进行了描述，但显而易见的是，可以在所附权利要求的范围内实施某些改变和变化。因此，所描述的实施例应当理解为示例性的而非限制性的，并且本发明不应局限于本文给定的细节，而是应当由所附权利要求及其等同设置的全部范围来限定。

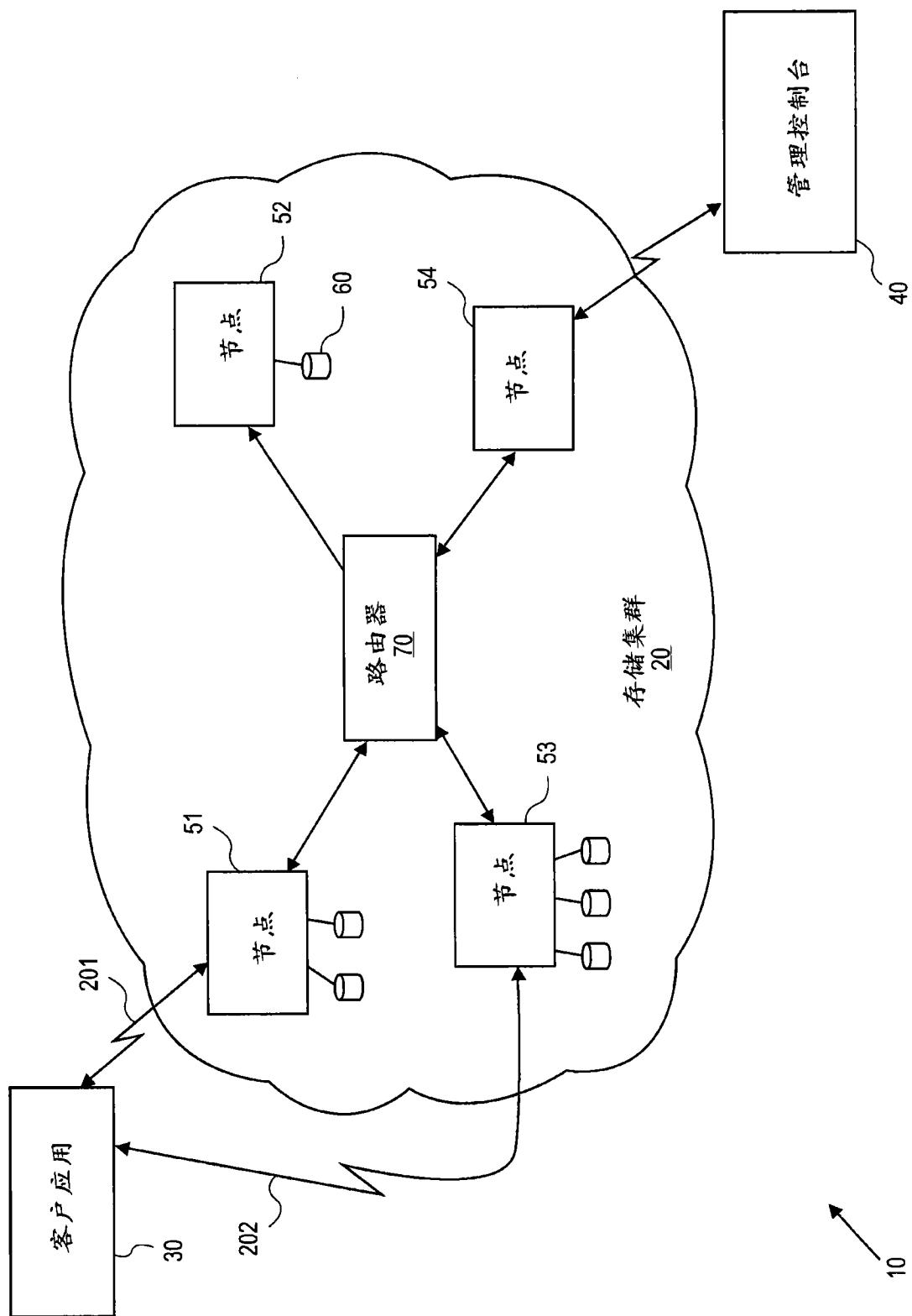


图 1

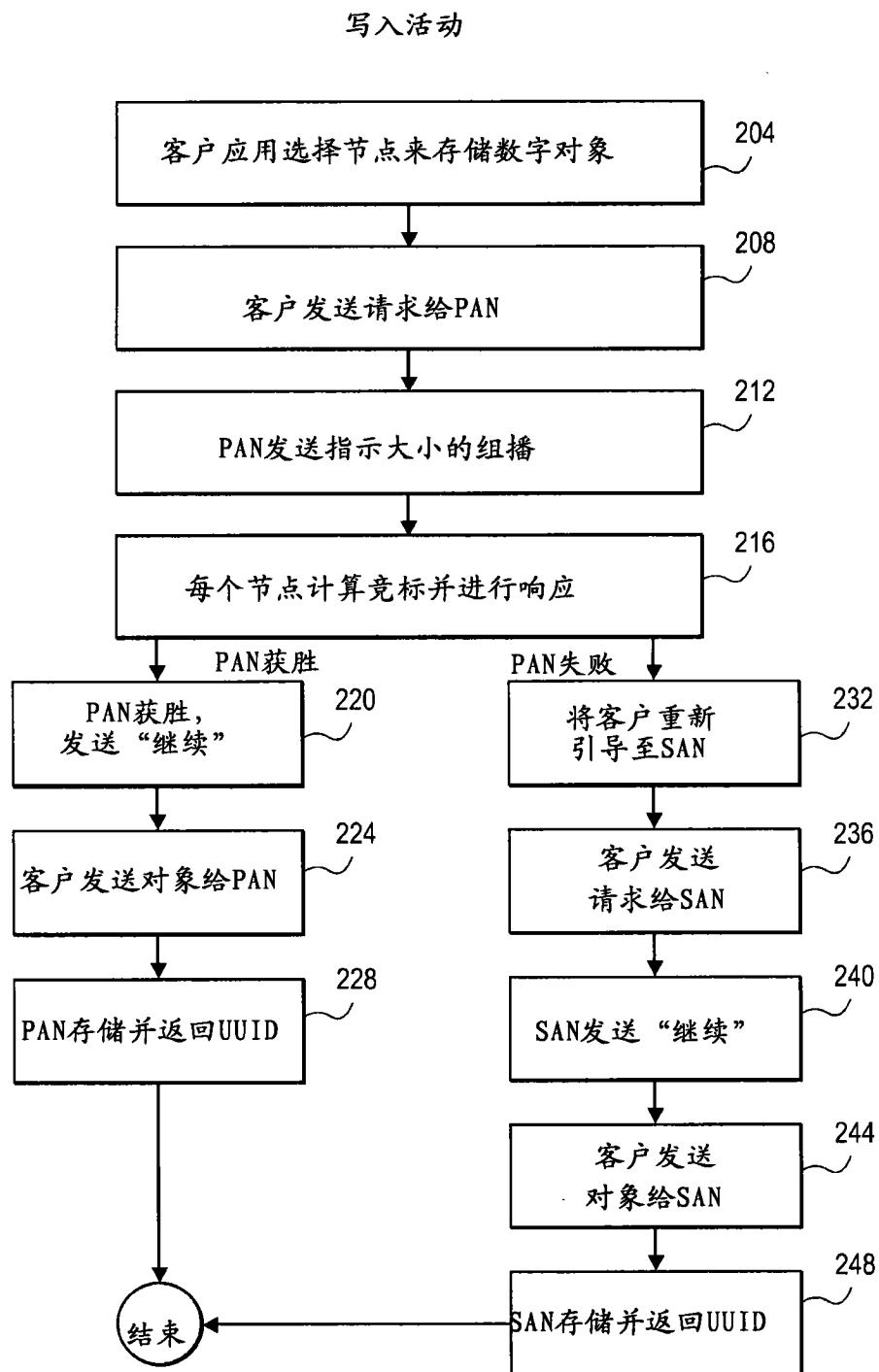


图 2

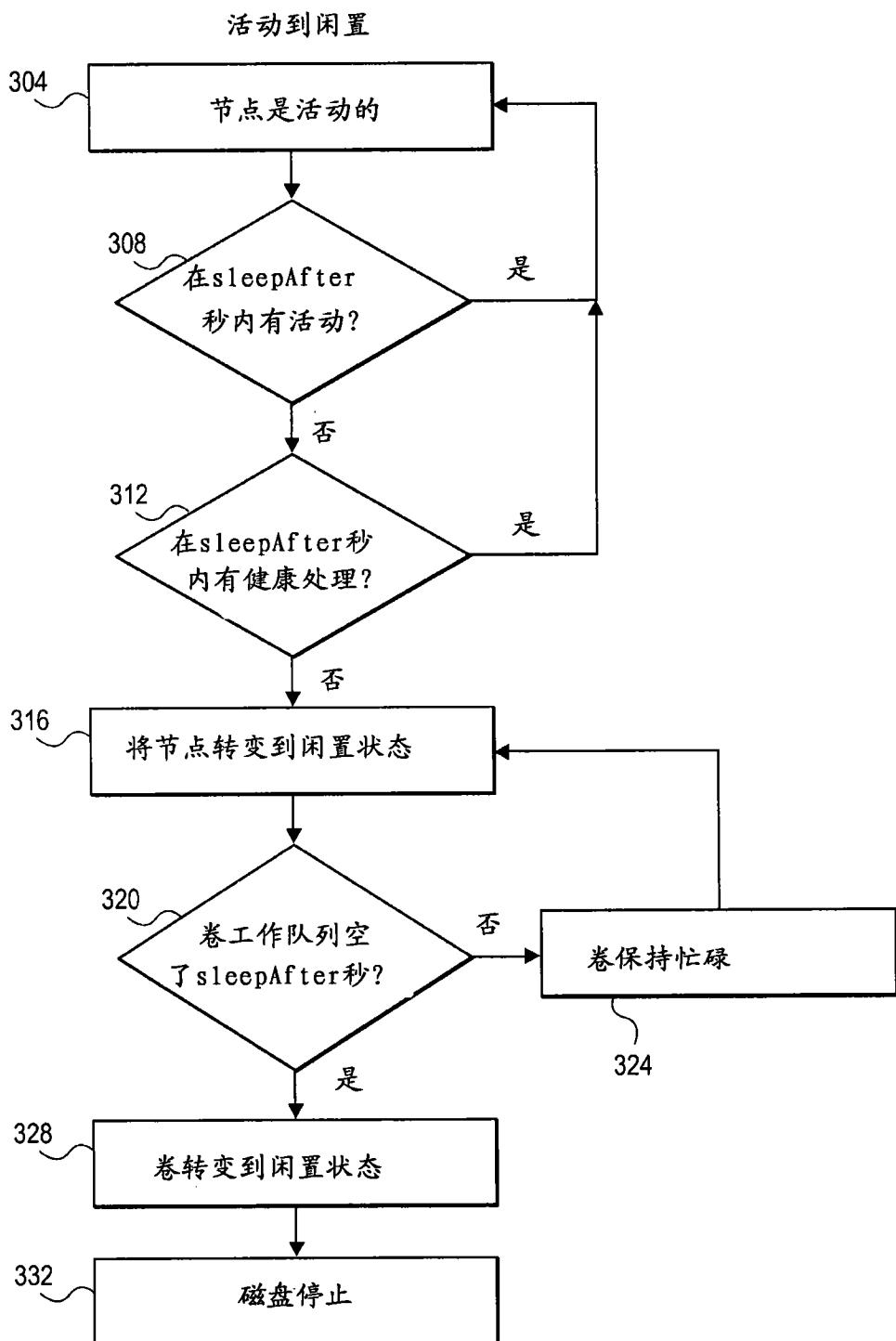


图 3

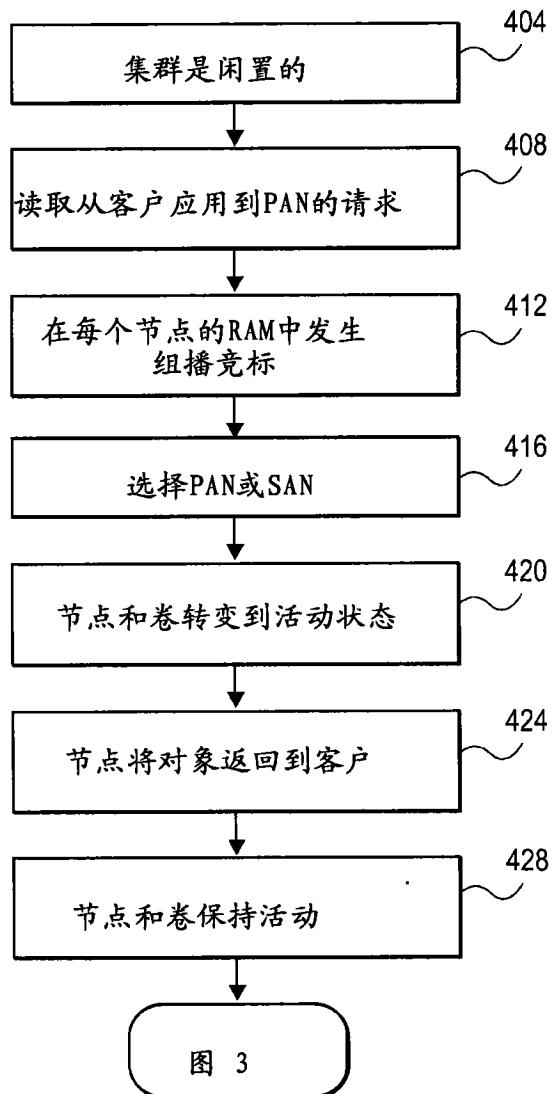


图 4

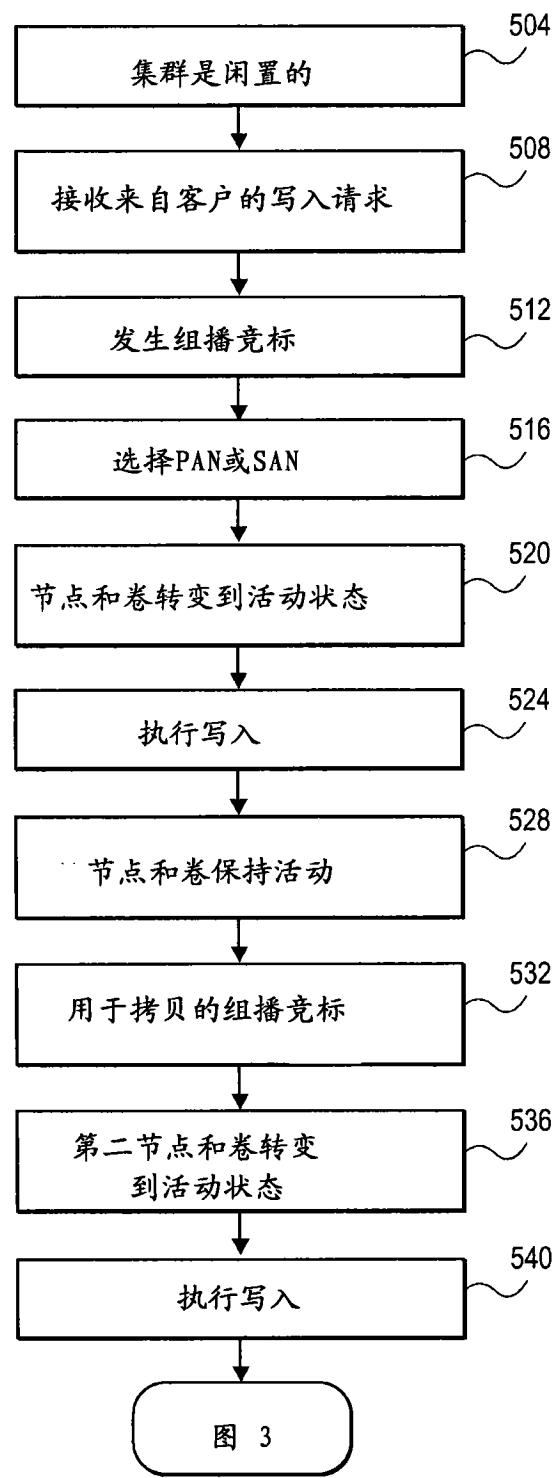


图 5

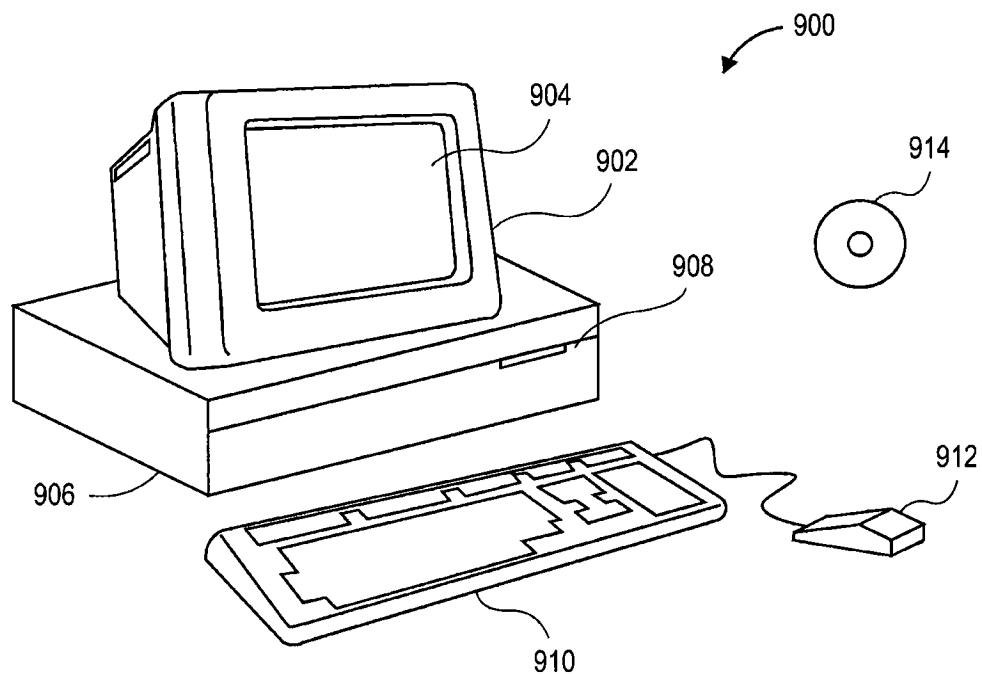


图 6A

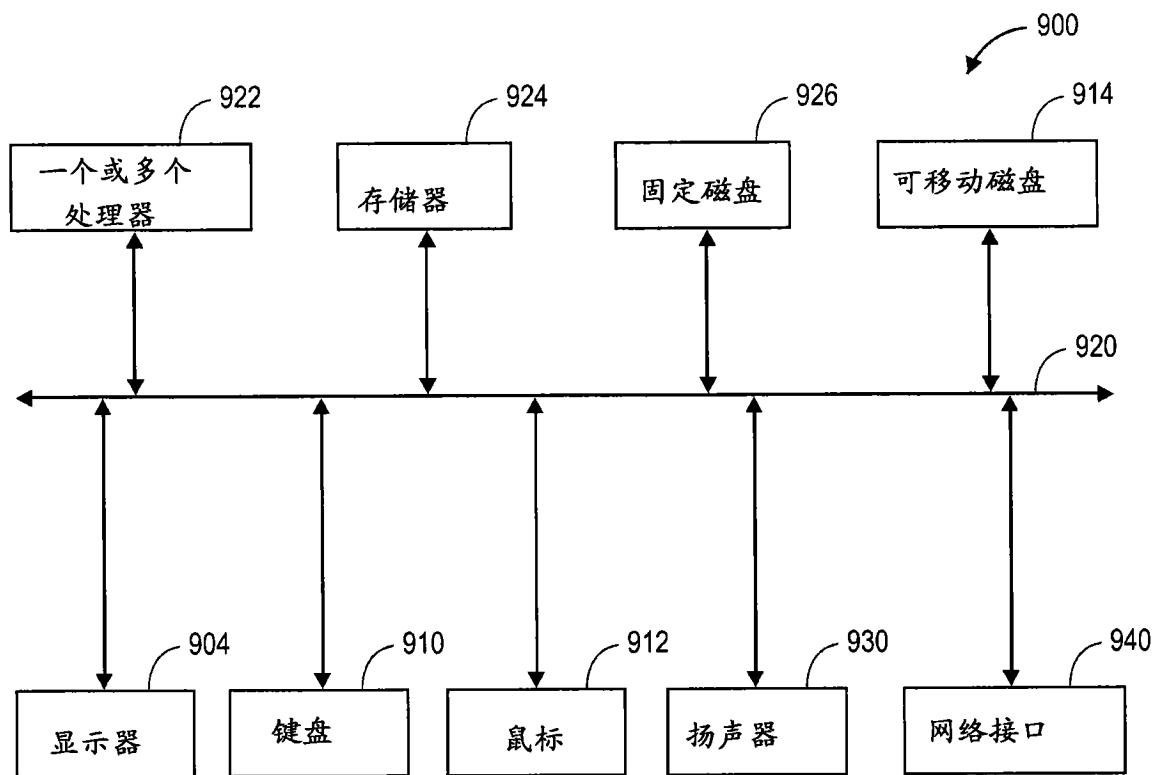


图 6B