[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 12/00 (2006.01)
G06F 17/30 (2006.01)
H04L 12/24 (2006.01)

[12] 发明专利说明书

专利号 ZL 200610101995.6

[45] 授权公告日 2008 年 6 月 11 日

[11] 授权公告号 CN 100394404C

[22] 申请日 2006.7.18

[21] 申请号 200610101995.6

[30] 优先权

[32] 2005. 8. 2 [33] US [31] 11/195,152

[73] 专利权人 国际商业机器公司

地址 美国纽约

[72] 发明人 文卡特斯瓦拉奥·尤尤里
克莱格·福尔梅·埃弗哈特
马拉哈尔·R·纳伊尼尼
罗西特·克里什纳·普拉萨德
森西尔·拉加拉姆

[56] 参考文献

US6535970B1 2003.3.18

CN1549981A 2004.11.24

US2005/0015354A1 2005.1.20

US2004/0098539A1 2004.5.20

US6862733B1 2005.3.1

审查员 郑宗玉

[74] 专利代理机构 中国国际贸易促进委员会专利商标事务所

代理人 李德山

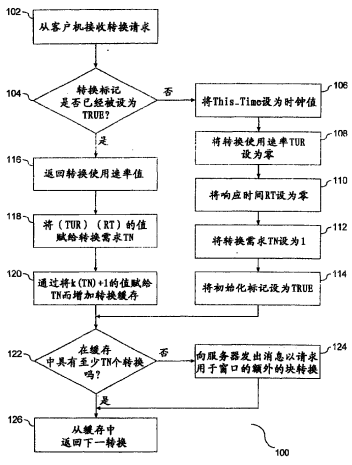
权利要求书 2 页 说明书 10 页 附图 5 页

[54] 发明名称

用于管理元数据的方法和系统

[57] 摘要

保持对在客户机－服务器系统中针对读写操作而由客户机使用的块地址缓存。 块地址被保持在缓存中，并且响应于对地址的请求而被流传输到客户机，以支持读或写操作。 响应于用于读写操作的地址的消耗，可以动态地调整保持在缓存中的地址数量。



1、一种用于管理块地址的方法，包括:

发送块地址请求;

接收响应于所述块地址请求的地址转换；并且

保持可用的块地址的缓存以供客户机使用，其中所述缓存的大小可根据块地址的历史使用来调整。

2、根据权利要求 1 所述的方法，其中

保持可用块地址的缓存的步骤包括以至少与所述客户机计算出的地址转换使用速率匹配的速率来发送对额外地址的请求。

3、根据权利要求 2 所述的方法，还包括根据所述客户机的所述使用速率以及服务器响应速率来适应性地修改所述缓存的最小尺寸。

4、根据权利要求 3 所述的方法，其中根据客户机对块地址的使用速率来修改所述缓存的尺寸，其中所述使用是从由读数据和写数据构成的组中选出的。

5、根据权利要求 1 所述的方法，还包括，在所述服务器创建额外地址的同时，所述客户机向所述创建的地址写数据。

6、一种用于管理块地址的系统，包括:

用于传送块地址请求的缓存管理器;

所述缓存管理器用于接收响应于所述块地址请求的被请求块地址；以及

与所述缓存管理器通信的缓存，该缓存用于保持由所述缓存管理器接收的所述块地址以供客户机使用，其中所述缓存的大小可根据块地址的历史使用来调整。

7、根据权利要求 6 所述的系统，其中所述缓存管理器用于以至少与所述客户机计算出的块使用速率匹配的速率来请求额外的地址。

8. 根据权利要求 7 所述的系统，其中所述缓存管理器用于根据所述客户机的所述块使用速率以及所述服务器的响应速率来修改所述缓存的最小尺寸。

9、根据权利要求 8 所述的系统，其中所述缓存管理器用于根据所述客户机对块地址的使用速率来修改所述缓存的尺寸，其中所述使用是从由读数据和写数据构成的组中选出的。

10、根据权利要求 6 到 9 中任一项所述的系统，其中所述缓存管理器支持客户机在接收额外地址的同时向所创建的地址进行写操作。

11、一种用于管理元数据分配的方法，包括:

接收元数据块地址请求；并且

响应于所述请求而将元数据块地址流传输到由客户机保持的缓存中，

其中所述缓存保持所述流传输的供客户机使用的块地址，其中所述缓存的大小可根据块地址的历史使用来调整。

用于管理元数据的方法和系统

技术领域

本发明涉及响应于读写操作而管理元数据。更具体地，提供了一种对共享文件系统中的用于读写操作的地址转换（translation）窗口进行保持的技术。

背景技术

图1是现有技术的分布式文件系统的框图（10），该系统包括服务器群（20），多个客户机（12）、（14）和（16），以及存储区域网络（30）。各个客户机基于数据网络（40）与服务器群（20）中的一个或多个服务器（22）、（24）和（26）进行通信。类似地，各个客户机（12）、（14）和（16），以及服务器群（20）中的各个服务器，与存储区域网络（30）进行通信。存储区域网络（30）包括仅包含有针对相关文件的数据块的多个共享磁盘（32）和（34）。类似地，服务器（22）、（24）和（26）对位于存储区域网络（30）的元数据存储空间（36）中的与相关文件的地址和属性有关的元数据进行管理。各个客户机可以访问存储在SAN（30）的文件数据空间（38）上的一个对象或多个对象，但是不可以访问元数据空间（36）。在打开SAN（30）中的存储介质上存在的文件对象的内容时，客户机与服务器之一进行联系以获取对象元数据和锁。典型地，元数据为客户机提供与文件有关的信息，诸如该文件的属性以及在存储设备上的位置。锁为客户机提供要打开文件以及读或写数据所必需的特权。服务器在SAN（30）的元数据空间中进行对于被请求文件的元数据信息的查询。服务器将授权的锁信息和文件元数据传送给发出请求的客户机，其中包括构成文件的所有数据块的地址。一旦客户机持有锁并且知道一个或多个数据块地址，该客户机就能够直接从附属于SAN（30）的共享存

储设备（32）或（34）访问文件的数据。系统（10）中组件（包括群中的服务器节点、客户机设备、以及存储介质）的数量仅是例示的数量。该系统可被扩大以包括另外的组件，类似地，该系统可被缩小以包括较少的组件。因此，图1所示的组件并不被解释为限制因素。

如图1所示，所例示的分布式文件系统将元数据和数据分离存储。在一个示例中，服务器群（20）中的服务器之一保存与共享对象有关的信息，包括存储器中的客户机可能访问的数据块的地址。为了读取共享对象，客户机从服务器获取文件的一个或多个数据块地址，然后从存储器读取给定的一个或多个数据块地址处的数据。类似地，当向共享对象进行写入时，客户机请求服务器为数据创建存储器块地址，然后请求所分配的随后要写入数据的存储器块地址。一种用于读取共享对象的已知方法使用连续的块分配方法。对于被连续读取的共享对象，客户机向服务器请求与第一段对应的数据块地址。当接收到块地址时，客户机读取该第一段中的数据。如果应用要求客户机在第一段以外进行读取，则客户机向服务器请求与下一段相对应的数据块地址，并且在接收到数据块地址时，客户机读取该下一段中的数据。类似地，对于连续的写操作，客户机请求服务器为第一段分配空间。当接收到用于该第一段的数据块地址时，客户机将数据写入该第一段中的数据块中。如果应用要求另外的段，则客户机向服务器发出消息以分配用于第二段的空间，并且当客户机接收到数据块地址时将数据写入该下一段的块中。连续块分配方法的读和写操作两者都要求客户机在读或写过程中针对共享对象中的连续地址向服务器请求数据块地址。客户机使用这些数据块地址将在对象上的读或写操作转换成在存储设备上的读或写操作。因为客户机记录有对象中的数据地址与存储设备中的数据地址之间的对应关系，所记录的从服务器获取的数据块地址称为地址转换（**address translation**），因为它们使客户机将对象相关的地址转换成存储设备中的地址。类似地，从客户机到服务器的针对数据块地址的请求有时称为地址转换请求。因此，如果初始分配请求没有提供足够数量的数据块地址，那么连续块分配技术就要求额外的客户

机－服务器通信。

如上所述，连续块分配技术响应于在最初返回的段中的数据块地址的不足，支持客户机与服务器的多次通信。这导致增加的带宽消耗，也称为增加的网络业务量。因此，需要这样的技术，即减轻针对每个读操作和每个写操作的用于块地址转换的客户机－服务器通信。这样的技术应该确保通过减少客户机－服务器的事务（transaction）数量来减轻网络业务量，以及确保减轻磁盘空间的浪费。

发明内容

本发明包括对读写操作中使用的块地址转换进行存储的窗口化（windowing）技术。

在本发明的一个方面，提供了一种用于管理块地址的方法。发送块地址请求，响应于该块地址请求而接收地址转换。对可用块地址的缓存进行保持以供客户机使用，其中所述缓存的大小可根据块地址的历史使用来调整。可根据针对读数据而由客户机使用块地址的速率来修改缓存的尺寸。

在本发明的另一方面，提供了一种用于管理块地址的系统。提供了一种用于传送块地址请求的缓存管理器。缓存管理器接收响应于该块地址请求的被请求的块地址。与缓存管理器通信的缓存用于保持缓存管理器所接收的块地址以供客户机使用，其中所述缓存的大小可根据块地址的历史使用来调整。

在本发明的再一方面，提供了一种具有计算机可读信号承载介质的产品。提供了在该介质中的用于发送块地址请求的指令，并且提供了在该介质中的用于接收响应于该地址请求的地址转换的指令。另外，提供了在该介质中的用于对可用数据块的缓存进行保持以供客户机使用的指令。

在本发明的再一方面，提供了一种对元数据的分配进行管理的方法。接收元数据块地址请求；并且响应于所述请求将元数据块地址流传输到由客户机所保持的缓存。该缓存保持所流传输的块地址以供客户机使用，其中所述缓存的大小可根据块地址的历史使用来调整。

本发明的其它特征和优点将根据以下结合附图对本发明的当前

优选实施例进行的详细说明而变得明了。

附图说明

图 1 是现有技术的分布式文件系统的框图。

图 2 是例示了根据本发明的优选实施例，如何将块地址保持在缓存中以用于对共享对象进行读写操作的流程图。

图 3 是例示了保持对地址转换请求的响应时间的估计的处理的流程图。

图 4 是例示了响应于地址转换请求而对地址转换的使用速率进行估计的处理的流程图。

图 5 是例示了示出缓存和缓存管理器的客户机的框图。

图 6 是例示了示出事务管理器的服务器的框图。

具体实施方式

概述

从客户机到服务器的块地址请求与读写操作异步发生。用于连续数据段的额外块地址被保持在尺寸可变的缓存中，并在读写操作期间可用。如果读操作需要额外的数据块地址，则将足够数量的块地址保持在缓存中。类似地，如果写操作需要分配额外的块以及所分配的块的地址，则将这样的地址保持在缓存中。将额外的块地址存储在缓存中，这减少了在读写操作期间的客户机－服务器通信。类似地，根据与实际的读和/或写操作相关联的当前和历史的行为模式，来动态地调整缓存的尺寸。通过基于这样的使用调整缓存的尺寸，也称为窗口，保持了对块地址需求的更加准确的估计。

技术细节

图 2 的流程图（100）示出了如何由客户机将块地址保持在缓存中，以解决在客户机－服务器的文件系统中对共享对象进行读写操作的效率。窗口是缓存为连续数据段所存储的块地址的数量。用于读写事务的块地址的集合也称作地址转换。因为该流程图集中于保持块地

址，所以可应用于读操作和写操作两者。如图所示，服务器接收来自客户机的块地址请求（102）。在接收到请求之后，进行测试，以确定转换初始化标记是否被设置为真（True）（104）。转换初始化标记用于确定是否已经建立了缓存。对步骤（104）的测试的假响应（false response）表示这是对块地址的第一次请求，并且需要初始化针对缓存的块地址数量的确定处理。该初始化处理包括：捕获时钟值，并将This_Time变量设置为该捕获值（106）；将转换使用速率（translation usage rate）变量TUR设置为零（108）；将响应时间变量RT设置为零（110）；将转换需求（translation need）变量TN设置为1（112）；并且将初始化标记设置为真（114）。步骤（106）、（108）、（110）、（112）、和（114）的集合提供了缓存的建立并将转换保持在缓存中。

对步骤（104）的测试的肯定响应表示已经建立了缓存以及相关联的地址窗口。该窗口是缓存为连续数据段存储的块地址的数量。返回转换使用速率TUR的当前值（116）。TUR是响应于地址转换请求而由客户机使用地址转换的频率。在一个实施例中，分别是，客户机可以使用缓存中的地址转换来读或写数据，而额外的地址由服务器提供或创建，并被传送到缓存以用于当前或将来的事务。对于读事务，客户机可以使用缓存中的可用地址转换来获取存储介质中的块地址以读取指定数据。在客户机进行读事务时，服务器基于TUR将额外的地址转换传送到缓存以确保缓存中有足够数量的地址可用，以针对当前事务以及将来的事务满足客户机的需要。对于写事务，客户机可以使用缓存中的可用地址转换来获取在存储介质中的可用于存储数据的块地址。在客户机进行写事务时，服务器基于TUR将额外的地址转换转发到缓存，以确保缓存中有足够数量的地址可用，以满足当前的写事务，以及将来的事务。以下详细说明的图4概述了如何确定转换使用速率。在步骤116之后，如下将乘积值赋给转换需求变量TN（118）：

$$TN \leftarrow (TUR) \times (RT)$$

转换需求变量TN表示由一个或多个客户机要求的块地址的数

量。在步骤（118）计算出的转换需求是基于客户机的使用速率（TUR）和服务器的响应速率（RT）而要存储在缓存中的块地址的最小数量。窗口的尺寸，即，待存储在缓存中的转换的数量，是通过如下将值赋给转换需求而增加的（120）：

TN←——k(TN) + 1

其中 k 是常数。在一个实施例中，将常数 k 的值选择为使得将块地址需求估计为至少与匹配于客户机计算出的块地址使用速率的速率一样大。可以基于由客户机针对读或写数据而使用地址的速率来动态地调整该常数的值。在步骤（120）或步骤（114）之后，进行测试（122）以确定缓存中的块地址的数量是否满足或者超过了在步骤（112）或步骤（120）所赋的 TN 值。对步骤（122）的测试的否定响应将导致向服务器发出消息以请求用于窗口的额外的块地址（124），如图 3 中详细所示。在步骤（124）之后、或者在对于步骤（122）的测试的否定响应之后，将块地址从块地址窗口返回给发出请求的呼叫方（126）。

注意到在以上的图 2 中，服务器对客户机对于要保持在缓存中的块地址的请求进行响应所需的时间被存储为变量 RT。图 3 是例示了用于计算 RT 变量的处理的流程图（150）。当在客户机与服务器之间开始事务时，捕获时钟的值，并将 Start_Time 变量设置为所捕获的值（152）。在步骤（152）捕获时钟值之后，向服务器发送消息以检取额外的块地址（154）。服务器对客户机的应答将包括用于该缓存的额外的块地址（156）。对于写事务，服务器也可以创建一些被返回给客户机以存储在缓存中的块地址。当客户机接收到块地址，即地址转换时，捕获时钟的值，并将变量 End_Time 设置为所捕获的值（158）。如下来计算客户机向服务器发送块地址请求与从服务器接收块地址之间的时间间隔：

Interval←——End_Time --Start_Time

在确定对服务器的响应时间的估计之前，进行测试以确定响应时间变量 RT 是否具有零值（162）。如果在图 2 的步骤 106 至 114 中系统刚被初始化，则响应时间变量 RT 将具有零值。对步骤（162）的测

试的肯定响应表示响应时间变量需求被初始化，并将使得将响应时间变量 RT 设置为在步骤（160）捕获的间隔值（164）。然而，对步骤（162）的测试的否定响应将使得如下来设置响应时间变量 RT（166）：

$$RT \leftarrow \alpha_2(Interval) + (1-\alpha_2)(RT)$$

其中，α_2 是 0 与 1 之间的常数。因此，保持对服务器的响应时间的估计的处理包括将块地址从缓存返回到发出请求的客户机。

另外，如以上图 2 中所示，发出请求的客户机使用地址转换的速率被作为变量 TUR 保持。图 4 是例示了计算 TUR 变量的处理的流程图（200）。存储在变量 This_Time 中的在最后一个地址转换开始时的时钟值被赋给变量 Last_Time（202），然后捕获时钟的当前值，并赋给变量 This_Time（204）。如下来计算由在步骤（202）和（204）捕获的时钟值所限定的时间间隔（206）：

$$Interval \leftarrow This\_Time - Last\_Time$$

在步骤（206）计算出时间间隔之后，进行测试以确定转换使用速率是否具有零值（208）。在图 2 的步骤 106 至 114，如果系统刚被初始化，则地址转换速率 TUR 将具有零值。如上所定义的，转换使用速率 TUR 是响应于地址转换请求而由客户机使用地址转换的频率。在步骤（208）的测试的肯定响应表示转换使用速率需求被初始化，并将导致将 TUR 变量设置为值（1/Interval）（210）。类似地，对步骤（208）的测试的否定响应表示转换使用速率已经被初始化。如下来设置转换使用速率（212）：

$$TUR \leftarrow \alpha_1(1/interval) + (1-\alpha_1)(TUR)$$

其中，α_1 是 0 与 1 之间的常数。因此，对保持在缓存中的块地址的使用速率的估计进行保持的处理包括将在最后一个地址转换的开始时的时钟值用作比较的基础。

用于保持块地址缓存的方法和系统基于地址的历史使用对保留在缓存中的块地址的数量进行动态地调整，这减少了网络业务量，并减少了与客户机－服务器通信相关联的等待时间。保持地址缓存以便由发出请求的客户机延迟使用的方法可以按照由客户机利用的工具

（tool）的形式被调用，以更快地从服务器传送块地址。图 5 是在系统中使用的客户机（305）的框图（300），示出了元数据分配工具的部件。如图所示，客户机（305）包括具有缓存管理器（312）和地址转换缓存（314）在内的存储器（310）。如以上所定义的，地址转换是用于读写事务的块地址的集合。当客户机（305）从服务器（未示出）接收到地址转换（316）时，将地址转换（316）存储在缓存（314）中。缓存管理器（312）保持优选数量的可供客户机（305）使用的地址转换，以减轻对来自客户机（305）的地址转换请求的传送，并减轻对客户机接收的地址转换的传送。将缓存管理器（312）设置为按照至少与客户机（305）计算出的地址使用速率匹配的速率从服务器（未示出）请求额外的块地址。地址转换（316）被存储在缓存（314）中，同时缓存管理器（312）响应于历史的和当前的地址转换请求而保持并动态地调整保存在缓存（314）中的转换的数量。尽管缓存管理器（312）被示为驻留在存储器（310）中，但是其应该并不限于软件部件。缓存管理器（312）也可以硬件部件来实现，并且驻留在存储器（310）的外部。

图 6 是在系统中使用的服务器（355）的框图（350），示出了元数据分配工具的部件。如图所示，服务器（355）包括存储器（358），该存储器（358）具有事务管理器部件（360），以便于针对为读或写事务预先创建的数据转发数据块的地址转换，并便于为写事务创建块地址。事务管理器（360）响应于从客户机（305）中的管理器（312）接收的元数据分配请求。事务管理器（360）被设置为便于响应于来自客户机的针对读写事务的块地址请求而创建并传送块地址。尽管事务管理器（360）被示为驻留在存储器（358）中，但是其应该并不限于软件部件。事务管理器（360）可以硬件部件来实现，并且驻留在存储器（358）的外部。与图 5 和图 6 中示出的部件相关地，该工具可以包括通过网络从客户机（305）传送到服务器（355）的块地址请求（365），以及对保持在缓存（314）中的地址转换（316）的返回，该地址转换（316）由客户机（305）用于读写数据。可以响应于当前的客户机请

求，或者将来的客户机请求来利用地址转换的缓存（314）。

在一个实施例中，如图 5 和图 6 所示，缓存管理器（312）和写事务管理器（360）可以是存储在计算机可读介质上的软件部件，只要该计算机可读介质包含有用作管理器的机器可读格式的数据。类似地，由客户机使用的可用块地址缓存还可以按照机器可读格式嵌入存储器中，以支持在客户机、服务器和存储介质之间的通信，该可用块地址是由服务器返回的供客户机延迟使用的。对于本说明书的目的，计算机可用、计算机可读、以及机器可读介质或格式可以是能够包含、存储、传送（communicate）、传播（propagate）、或传输（transport）程序的任何装置，该程序被指令执行系统、装置或设备使用或与其相关。因此，元数据块地址请求、返回的块地址、以及可用块地址缓存可以全部是计算机系统中的硬件元件的形式，或者计算机可读格式的软件元件，或者是软件和硬件的组合。

优于现有技术的优点

当客户机执行读或写操作时，客户机向服务器发送针对块地址转换的请求。这些转换被容纳在由客户机保持的缓存中，并且可方便地用于客户机当前或以后的操作。在缓存窗口中保持限定数量的块地址，这样通过减少客户机－服务器通信而减轻了网络业务量。当块地址可用时，客户机可以直接从缓存获得块地址，而不是针对各个事务从服务器请求块地址。另外，可根据块地址的历史使用来调整保持在缓存中的块地址的数量。这确保了缓存中的块地址的数量既不会用不了，也不会不够用。另外，利用各个客户机－服务器事务、或利用各个读和/或写操作、或利用这两者来跟踪历史使用，从而使得可以动态地对缓存的尺寸，也称为窗口进行调整。因此，缓存用作用于改善关于读和写操作的通信效率的工具。

另选实施例

应该理解，尽管为了例示的目的，在此对本发明的具体实施例进行了说明，但是在不脱离本发明的精神和范围的情况下，可以作出各种修改。具体地，在步骤（120）赋予的用于增加块地址缓存的窗口的

常数可以被修改，以根据系统需求增加或减少保存在缓存中的块地址的数量。类似地，常数α_1和α_2可以根据一初始值被重新设置，以进一步对保持在缓存中的块地址的数量进行调整。在一个实施例中，常数α_1和α_2优选地被设置为0与1之间的值，一般被设置为值1/8。类似地，可以开发出其它的算法，用于根据使用来估计出客户机将来使用的有用窗口尺寸。另外，算法可以被复制，一次用于读操作、并且一次由写操作使用。类似地，在流程图例示中概括的功能可被有目的地分隔成多个单独的处理或执行线程，以提高并行性。在一个实施例中，单独的处理或执行线程将利用块地址来填充缓存窗口，而客户机的读或写操作利用已经在手边的块地址来进行。因此，本发明的保护范围仅由以下的权利要求及其等同物来限定。

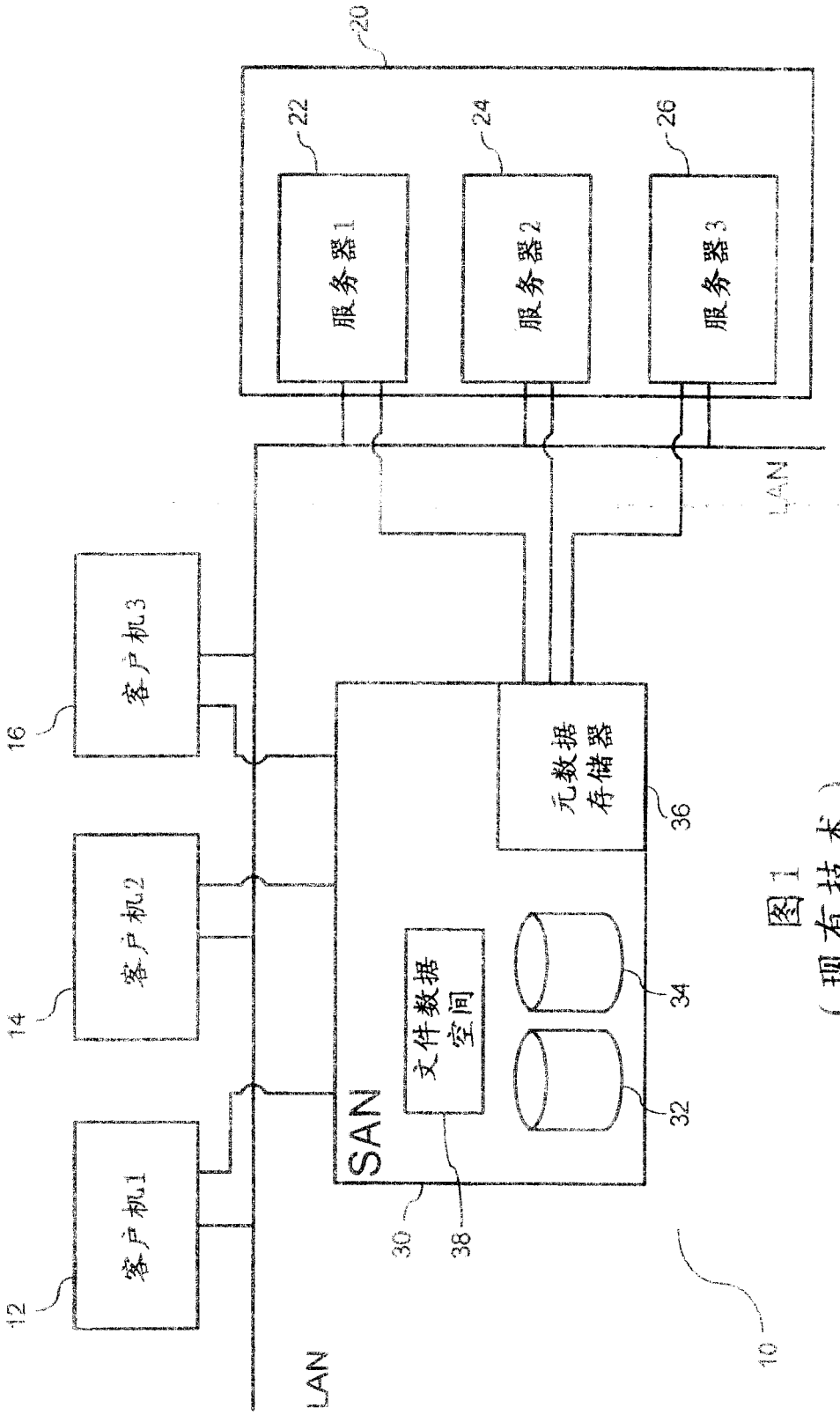

20
22
24
26
服务器1
服务器2
服务器3
LAN
16
客户机3
14
客户机2
12
客户机1
LAN
SAN
元数据存储器
36
文件数据空间
34
32
30
38
10

图1
（现有技术）

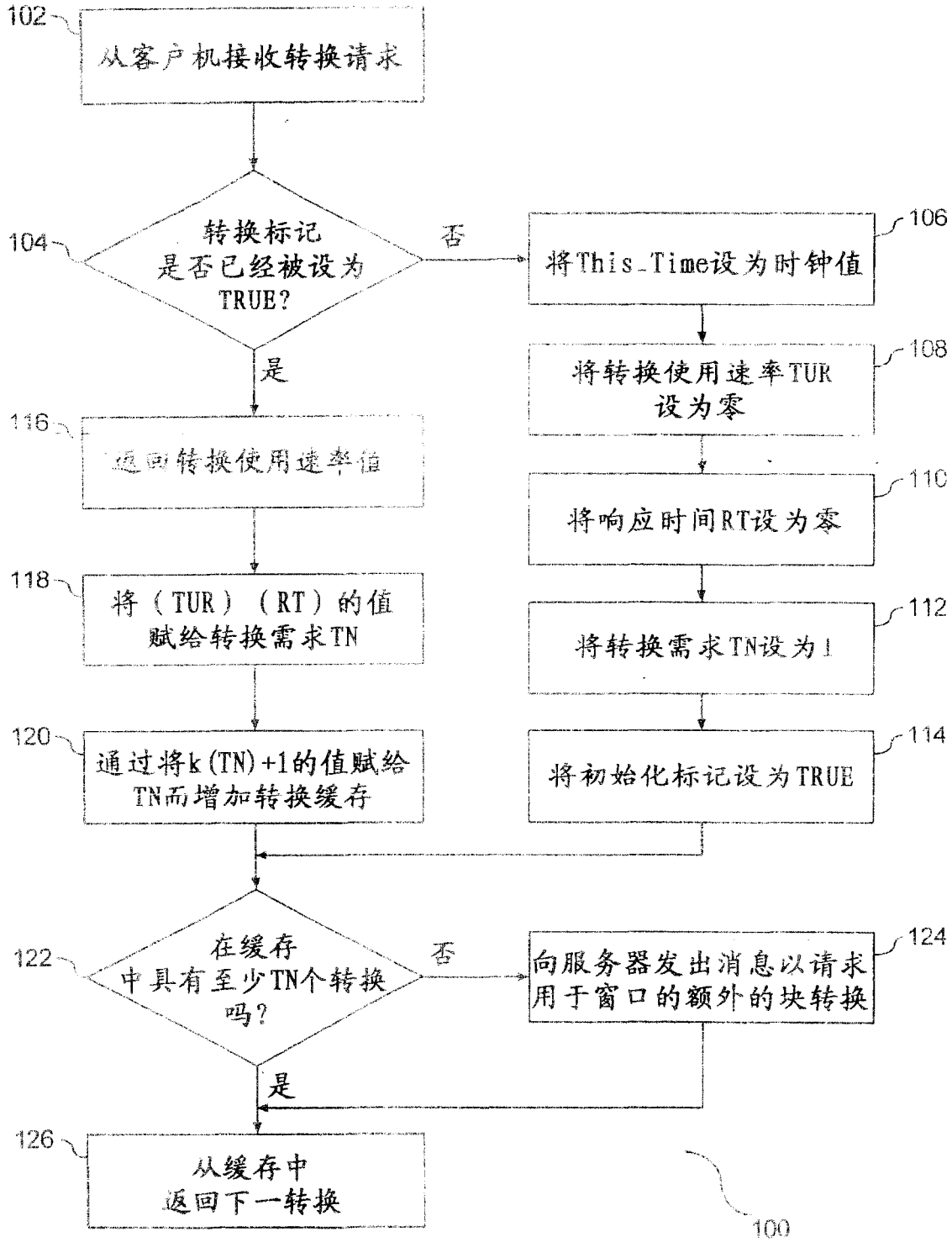

102
从客户机接收转换请求
104
转换标记
是否已经被设为
TRUE?
否
是
106
将This_Time设为时钟值
108
将转换使用速率TUR
设为零
110
将响应时间RT设为零
112
将转换需求TN设为1
114
将初始化标记设为TRUE
116
返回转换使用速率值
118
将（TUR）（RT）的值
赋给转换需求TN
120
通过将k(TN)+1的值赋给
TN而增加转换缓存
122
在缓存
中具有至少TN个转换
吗?
否
124
向服务器发出消息以请求
用于窗口的额外的块转换
是
126
从缓存中
返回下一转换
100

图2

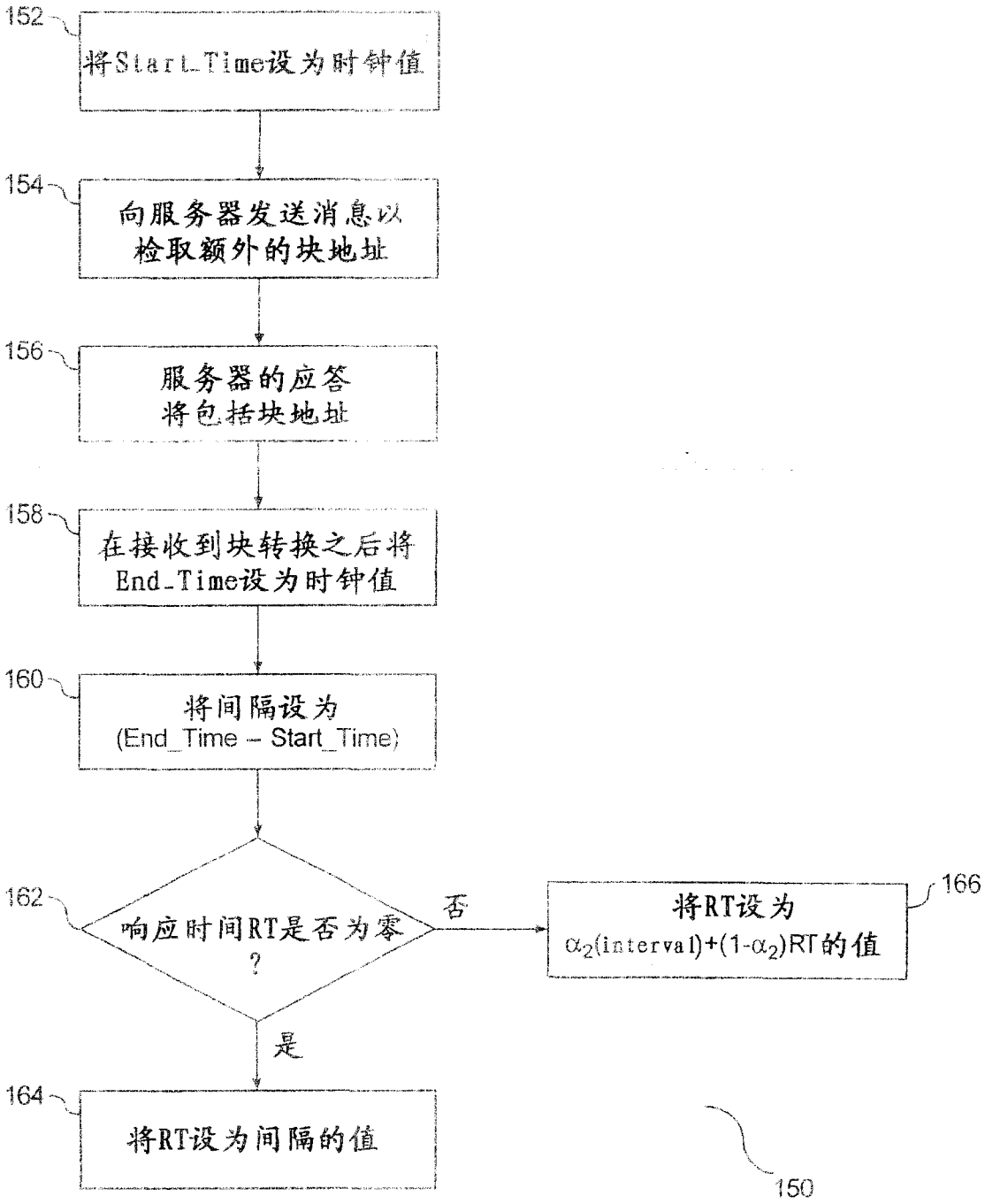

152
将Start_Time设为时钟值
154
向服务器发送消息以
检取额外的块地址
156
服务器的应答
将包括块地址
158
在接收到块转换之后将
End_Time设为时钟值
160
将间隔设为
(End_Time – Start_Time)
162
响应时间RT是否为零
?
否
166
将RT设为
α_2(interval)+(1-α_2)RT的值
是
164
将RT设为间隔的值
150

图3

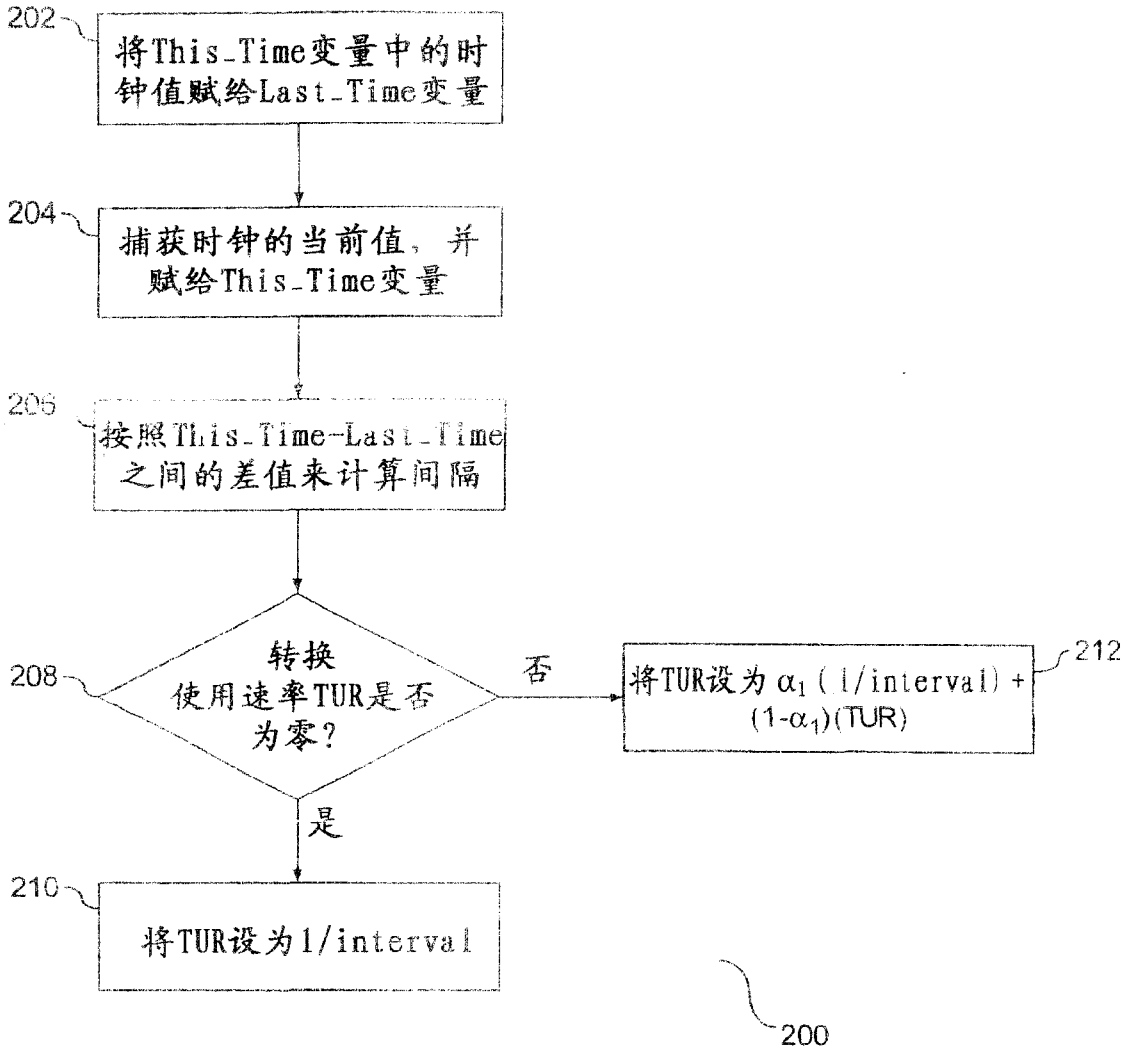

202
将This_Time变量中的时钟值赋给Last_Time变量
204
捕获时钟的当前值，并赋给This_Time变量
206
按照This_Time-Last_Time之间的差值来计算间隔
208
转换使用速率TUR是否为零?
否
212
将TUR设为 α_1 (1/interval) + (1-α_1)(TUR)
是
210
将TUR设为1/interval
200

图4

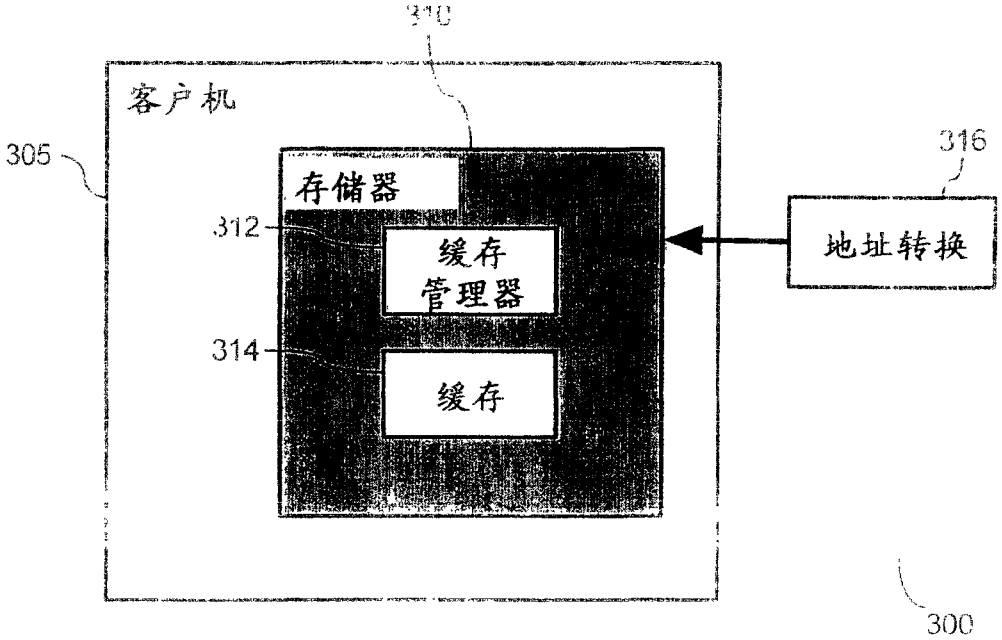

客户机
305
存储器
312
缓存
管理器
314
缓存
316
地址转换
300

图 5

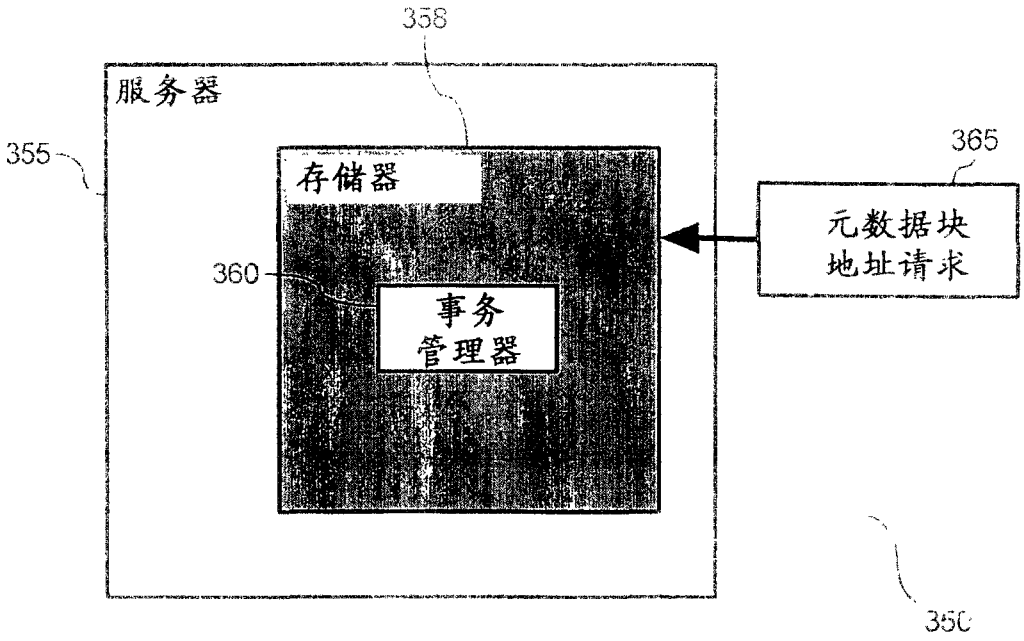

服务器
355
存储器
360
事务
管理器
365
元数据块
地址请求
350

图 6