[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
H04L 29/06 (2006.01)

[12] 发明专利申请公开说明书

[21] 申请号 200510086076.1

[43] 公开日 2006 年 1 月 25 日

[11] 公开号 CN 1725758A

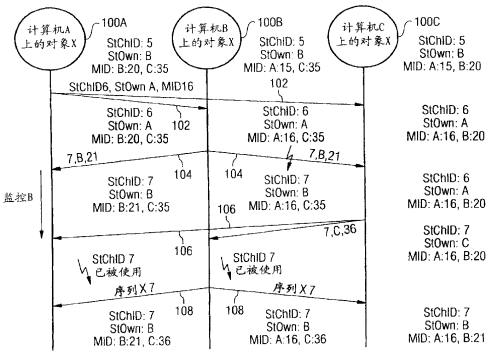
[22] 申请日 2005.7.19
[21] 申请号 200510086076.1
[30] 优先权
[32] 2004. 7.19 [33] EP [31] 04017035.9
[71] 申请人 西门子公司
地址 德国慕尼黑
[72] 发明人 M·耶卡尔 G·施赖伯
C·施泰因布吕克 S·冯德黑德
[74] 专利代理机构 中国专利代理(香港)有限公司
代理人 程天正 张志醒
权利要求书 3 页 说明书 8 页 附图 2 页

[54] 发明名称

用于使分布式系统同步的方法

[57] 摘要

本发明涉及使分布式系统的组件同步的方法。系统状态由至少一个在所有组件中设定的对象来代表。借助于状态转换消息用信令向所有其他组件发送一个组件中的对象状态的转换。通过每个其他组件检查用信令发送状态转换的本地有效性，在本地有效的状态转换时更新这些组件中的对象的状态，而在本地无效的状态转换时确定具有至少向具有对象无效状态的组件发送的对象有效状态的组件，然后更新这些组件中的对象的状态。本发明涉及分布式系统，配置该分布式系统的组件以用于执行本发明。跟已知的分布式系统相比，本发明为了传输用于维持或者重新建立系统的同步性而必需的状态转换消息，可以使用不可靠的并从而快速的传输方法，例如 UDP 组播消息。



1. 用于使分布式系统的组件（A、B、C、D）同步的方法，其中，系统状态通过至少一个在所有组件（A、B、C、D）中设定的对象（X）来代表，其中借助于状态转换消息（102、104、106）将在组件之一中的对象（X）的状态转换用信令发送给所有其他组件，随后通过每个其他组件来检查用信令发送的状态转换的本地有效性，其中在本地有效的状态转换时，更新这些组件中对象的状态，并且在本地无效的状态转换时，确定具有至少向具有对象（X）无效状态的组件所发送的对象（X）有效状态的组件，然后更新在这些组件中的对象的状态。

2. 按照权利要求 1 所述的方法，其特征在于，借助于不可靠的组播传输所述状态转换消息（102、104、106）。

3. 按照权利要求 1 或者 2 之一所述的方法，其特征在于，将发送组件的至少一个连续的状态转换消息标志（StChID）、进行状态转换的组件的标志（StOwn）和一个本地消息标志（MID）分配给每个状态转换消息（102、104、106）。

4. 按照权利要求 3 所述的方法，其特征在于，

－所有接收一个状态转换消息的组件将与在相应组件上接收或者发送的最后的状态转换消息的状态转换消息标志（StChID）相对应的相应本地状态转换消息标志（StChID）与所接收的状态转换消息（102、104、106）的状态转换消息标志（StChID）进行比较，以便检查用信令发送的状态转换的本地有效性，和

－在不相同时，确定决定状态的组件，通过该组件重新确定所有其他组件的对象（X）的状态。

5. 按照权利要求 4 所述的方法，其特征在于，从进行状态转换的组件的标志（StOwn）、状态转换消息标志（StChID）和分配给所述组件的优先权中确定所述决定状态的组件。

6. 按照权利要求 4 或者 5 之一所述的方法，其特征在于，通过从上述比较中确定对象（X）的本地状态与当前状态不对应的组件，请求在发送所述状态转换消息的组件中的所有对象（X）的副本。

7. 由组件（A、B、C、D）组成的分布式系统，所述系统的状态通过至少一个在所有组件（A、B、C、D）中具有的对象（X）代表，其中每个组件具有：

－用于借助于状态转换消息（102、104、106）用信令向所有其他组件发送对象（X）的状态转换的装置；

－用于检查用信令发送的状态转换的本地有效性的装置；

－用于响应于已接收了本地有效的状态转换消息（102、104、106）而更新对象（X）的状态的装置；

－用于确定具有对象（X）有效状态的组件的装置，以及用于响应于已接收了本地无效的状态转换消息而由该组件接收对象（X）的状态的装置。

8. 按照权利要求 7 所述的系统，其中用于用信令发送对象（X）的状态转换的装置包括用于借助于不可靠的组播来传输所述状态转换消息（102、104、106）的装置。

9. 按照权利要求 7 或者 8 之一所述的系统，其中用信令发送对象（X）的状态转换的装置包括用于产生状态转换消息（102、104、106）的装置，其中至少一个连续的状态转换消息标志（StChID）、进行状态转换的组件的标志（StOwn）和发送组件的本地消息识别（MID）被分配给所述用于产生状态转换消息的装置。

10. 按照权利要求 9 所述的系统，其组件（A、B、C、D）附加地包括：

－用于存储与在相应组件中接收或者发送的最后状态转换消息的状态转换消息标志（StChID）对应的相应本地状态转换消息标志（StChID）的装置；

－用于将所述本地状态转换消息标志（StChID）与所接收的状态转换消息（102、104、106）的状态转换消息标志（StChID）进行比较的装置，以便检查信令化的状态转换的本地有效性；和

－用于在不相同时确定决定状态的组件的装置，以及用于通过所确定的组件重新确定所有其他组件的对象（X）的状态的装置。

11. 按照权利要求 10 所述的系统，其中用于在不相同时确定决定状态的组件的所述装置包括用于分析下列参数的装置：进行状态转换的组件的标志（StOwn）、状态转换消息标志（StChID）和分配给所述组件的优先权。

12. 按照权利要求 9 或者 10 之一所述的系统，其组件（A、B、C、D）另外还具有装置用于响应于已确定了本地状态与对象（X）的系统

范围的状态之间的误差而请求在发送状态转换消息的组件中的所有对象（X）的副本。

13. 按照权利要求 7 至 12 之一所述的系统，其中所述对象（X）代表所述组件之一，并且对象（X）的这种代表特性由一附加的属性表示。

用于使分布式系统同步的方法

技术领域

本发明涉及使分布式系统同步，并尤其涉及分配数据以及访问该分布式系统中的资源。

背景技术

为了完成复杂的计算技术任务，和/或为了在数据处理设备中通过冗余度实现安全，经常使用如下设备，其由多数单个自动装置或者计算机或者计算机系统组成，但是对设备的用户表现透明如同单个设备。也称这种设备为分布式系统，其中例如过程或者措施（如存储冗余度或者负载平衡或者负载分配）对用户来说优选地被形成为透明、即看不到的。分布式系统与网络的区别例如在于，在网络中，网络的单个计算机对用户表现为分离的实体，其中在网络中，存储冗余度或者负载平衡对用户来说被形成为经常不是透明的、即可看到的，并且经常甚至需要用户相互配合。

在分布式系统中，必须在单个计算机或者系统之间进行数据分配。对该数据分配，已知多种可能性。对此，通常可以区分为一方面相对慢的、交易可靠的数据传输，另一方面较快的、相对不可靠的具有微小耗费的分配。

如果在交易可靠的数据传输情况下交易可靠延伸到用户接口，那么可以在每个时刻保证完全的数据一致性，其方式是，例如只有在系统中已经可靠地分配了输入时，才通过用户获得反馈信息。但是，尤其在必须将数据分配给许多机器的情况下，这是具有高通信部分的相对费时的过程。当通常不期望长反应时间时，具有必须避免长反应时间的一系列应用、例如安全临界应用。另外，实现交易可靠的系统耗费巨大且价格昂贵。

另一方面，如果偏离完全交易可靠的原则，其中例如当可能将输入转移给仅一个其他机器或者其他系统时就已经产生反馈信息，那么在系统中存在不一致的状态的危险。例如如果由于连接问题出现系统分离，或者恰恰已经接收输入数据的那些机器或者部分系统出现故障，则这些不一致的状态可能引起数据损耗。因此，可能已经用信令发送它的输入的处理给用户，而在系统中丢失该输入，但是这特别不

允许出现在安全临界环境中。

把引起的紧急呼叫看作例子，例如通过指示器向发出紧急呼叫的用户证实该成功发出的、但最终不再被继续处理的紧急呼叫。

发明内容

因此，本发明的任务在于，给出分布式系统的可选的传输方法以及可选的分布式系统，由此，以较小的损耗来达到完全的交易可靠的可靠性。

该任务通过用于使分布式系统的组件同步的方法来解决，因此，系统状态通过至少一个在所有组件中设定的对象来代表，其中借助于状态转换消息用信令向所有其他组件发送组件之一中的对象的状态的转换，此后，通过每个其他组件检查用信令发送的状态转换的本地有效性，其中在本地有效的状态转换时，更新这些组件中的对象的状态，而在本地无效的状态转换时，确定具有至少向具有对象的无效状态的组件发送的对象有效状态的组件，然后更新在这些组件中的对象的状态。

另外，本发明涉及分布式系统，配置所述分布式系统的组件以用于执行本发明方法。

与已知的分布式系统相比，本发明提供了如下优点，即为了传输用于保持或者重新建立系统同步性所必需的状态转换消息，可以使用不可靠的并从而快速的传输方法，例如 UDP/IP（用户数据报协议/互联网协议）组播消息。这里假定，在通常情况下，这些消息也到达它们的目的地，并且系统单独地通过这些消息来保持同步。当然，如果出现干扰，那么每个组件最晚利用下一次接收的状态转换消息或者在消息传输中的较长暂停时通过所应用的监控机制能够，确定同步的（可能本地的）中断，并在具有正确状态的组件上请求该正确状态。在此，系统状态通过一个或者多个对象来完整地描绘。在此，多个适当受限的对象给出下列优点，即在对象之一的状态转换时出现的数据量较小。

换句话说，本发明实现一种弱耦合的分布式系统，其中没有受到干扰的正常运行比在完全交易可靠的系统中运行的要快，而同时保证完全交易可靠的系统的安全性。

这里，按照本发明，没有必要维持系统状态的信息或者诸如服务

器或者数据库等所选中央组件中的对象的状态信息，其中所述组件将构成所谓的单点故障（Single Point of Failure）。更确切地说，在故障情况下，确定具有有效状态的组件，所述有效状态然后被应用于具有无效对象状态的组件。按照本发明，总是通过中央组件限定的对系统的可供使用性产生消极影响的“单点故障”是不必要的。

本发明同样提供一种用于解决对专用资源的竞争访问的机制。如果两个或者多个实例同时或者时间相近地管理或者试图管理专用资源，那么在进行这种竞争访问时，调整一致的或者同步的系统状态是必要的。这里也可以通过本发明阻止或者消除不一致的状态。

在许多应用情况下，本发明分布式系统优选地可以代替用于分布式工作（如 CORBA）的特别的数据库以及传输机制。这种系统也出现暂时网络分离而后来又重新归并（分离的部分网络）的环境中具有优点。这里，系统（通常对用户来说是不可觉察的）再次处于共有的状态。

附图说明

下面，借助于附图在实施例中对本发明进行更详细的描述。图 1 和 2 示出在具有与处理传输错误有关的多个组件的弱耦合系统中与状态相关的通信的运行简图。

具体实施方式

图 1 示出与对象 X 相关的通信过程的简图，所述对象 X 在弱耦合系统的三个组件 A、B、C 上形成映像。通过圆 100A、100B、100C 分别针对三个计算机或者计算机系统 A、B、C 示出这些对象映像。

假设所有三个计算机 A、B、C 首先已经为对象 X 存储了相同的状态，因此也关于对象 X 同步地工作。该状态的特征由所谓的状态占有者（StOwn）和上一次的状态转换消息标志（StChID）来表征。该两个参数被存储在所有计算机中，并且只要系统同步地工作，那么系统范围内该两个参数相同。另外，在每个计算机中还存储所有其他计算机的其他消息标志（MID），也即在计算机 A 中存储有计算机 B 和 C 的其他消息标志，在计算机 B 中存储有计算机 A 和 C 的其他消息标志，和在计算机 C 中存储有计算机 A 和 B 的其他消息标志。

将计算机称为状态占有者，在所述计算机中已经发生了上一次的状态转换，随后该计算机借助于状态转换消息已将该状态转换通信给

其他计算机。在该情况下有：StOwn＝B，即计算机 B 是最后状态转换的状态占有者，其有标志 StChID＝5。由 A 发送的上一次的消息具有 MID 15，由 B 发送的上一次的消息具有 MID 20，和由 C 发送的上一次的消息具有 MID 35，由此得出图 1 所示的、关于与状态转换消息相关联的标志的初始状态。

基于状态转换，计算机 A 接下来向所有其他组件、即向计算机 B 和 C 传输具有标志 StChID＝6（旧 StChID 加 1）的状态转换消息 102。状态占有者从现在起是计算机 A，因为状态转换由所述计算机 A 开始。A 的 MID 也被增值，并且与状态转换消息一起被传输，它从现在起为 16。

所有其他组件正确地接收消息 102。将标志 StChID 和 MID 与本地存储的值进行比较，并且确定，在 B 和 C 处迄今的本地状态是有效的，因为所接收的 StChID 和 MID 的值正好对应于增加 1 的本地值。另外从 A 已经使用了正确的 StChID 和 MID 的事实中得出，该状态在计算机 A 处也有效。随后，计算机 B 和 C 更新对象 X 的状态，该状态然后再次对所有计算机都是相同的，其参数为 StChID＝6，StOwn＝A 和 MID A: 16，B: 20，C: 35。

一般来讲，在上述无干扰工作时，该系统表现如下：

原则上在每个参与的计算机上都产生代表一部分系统状态的所有对象。因此，整体上对所有具有系统状态的对象进行复制。经组播消息发送状态的每次变化，并且从而由所有组件或者计算机 A、B、C 接收。对象本地参数 StChID、参数 StOwn 和计算机本地参数或者组件本地参数 MID 被添加到这种消息上。

利用参数 StChID 可以识别由于同时转换状态或者暂时分离网络引起的冲突。借助于计算机本地参数 MID 可以识别暂时的网络分离。同时，一般通过主动的 Ping 机制在参与的计算机之间监控该参数，其中可以如此配置 Ping 机制，使得如果没有收到该计算机的其他消息，那么只交换 Ping。如果没有监控机制通知错误，那么因此通过简单发送（组播）消息来结束状态转换的传播。

再次根据图 1，基于状态转换，计算机 B 向所有其他组件、即向计算机 A 和 C 传输具有标志 StChID＝7（旧的 StChID 加 1）的状态转换消息 104。从现在起状态占有者为计算机 B，因为状态转换以该计

算机为起点。B 的 MID 也被增值，并且和状态转换消息被一起传输，它从现在起为 21。

消息 104 由 A 正确接收。计算机 A 重新进行上述检查，并且最终更新本地状态，于是，该本地状态由参数 StChID = 7，StOwn = B 和 MID A: 16，B: 21，C: 35 来表征。

由于通信干扰，计算机 C 没有或者没有正确接收消息 104。计算机 C 不执行动作，并且保持在迄今有效的状态中，该状态（同上）由参数 StChID = 6，StOwn = A 和 MID A: 16，B: 20，C: 35 来表征。

分布式系统从该时刻开始不再是同步的，但这不可直接地被确定。在图 1 的例子中，借助于下一次的状态转换消息进行错误识别，但是也可以通过所提及的 Ping 机制进行。通过 B 利用上一次的有效 MID 向 A 和 C 发送 Ping，可以通过 Ping 进行错误识别，随后 C 确定偏差，并且从而识别本地状态为无效，随后在图 2 的意义上进行错误处理。

如果在系统中借助于下一次的状态转换消息在系统中进行错误识别，那么可以区别两种情况：C 发送状态转换消息，而 A 和 B 识别该问题（图 1 所示，在下面进行说明），或者 A 或者 B 发送状态转换消息，而 C 识别该问题（关于图 2 进行表示和说明）。

基于状态转换，计算机 C 向所有其他组件、即向计算机 A 和 B 发送状态转换消息 106。因为在计算机 C 中的对象 X 的本地状态与当前状态不对应，所以 C 使用（从 A 和 B 的观点看）过时的状态转换消息 StChID = 7。C 的状态被 A 和 B 识别为无效，因为 C 期望 StChID = 8。

如果错误被识别，那么冲突要被解决。在此，借助于参数 StOwn、StChID 或/和可预定的优先权确定，通过哪个组件决定真实状态。在具有相同的权利时，借助于通常已知的可与实际状态占有者相比较的特性的最小值（例如借助于网络地址）确定组件。因为整体上存在为决定需要的所有数据，所以作出决定不需要附加的通信。

在图 1 的例子中假设，参数 StOwn 对确定支配真实状态的组件起决定作用，即选择 B 作为出现上一次的状态转换（具有 StChID=7）的组件。完整的对象由该所选择的组件至少被传输到具有无效本地状态的组件、也即计算机 C（步骤 108），并且也可以被并列地传输到所有其他的组件。借助于组播（序列 X7）连续地传输具有状态 StChID

=7 的对象 X。如果只给具有无效本地状态的组件重新提供对象 X，那么替代组播也可以选择交易可靠的传输，以便保证完全无误地传输对象映像。

在进行这种错误消除之后，所有组件 A、B、C 再次具有关于对象 X 的统一的状态，其特征在于，参数 StChID=7，StOwn=B 和 MID A: 16，B: 21，C: 36。

在图 2 中，以与图 1 中相同的情形（A 和 B 具有 StChID=7，由于错误，所以 C 具有 StChID=6）为出发点来描述确定组件 C 处的不一致性的情况，其中通过并列事件消除错误需要比图 1 的例子中更多的步骤。

组件 B 在步骤 104 中已经传输了新的状态（StChID=7）之后，在 A 中进行状态转换，该状态转换由 A 向其他组件、即向计算机 B 和 C 传输（步骤 206）。此外，使用状态转换消息参数 StChID=8，StOwn=A 和 MID=17。在所接收的状态转换消息中获得 StChID=8 时，C 确定：C 没有接收具有 StChID=7 的状态转换消息，并从而本地状态在 C 处无效。所以 C 通过广播（即向包括为了简化起见而迄今没有详细考察的组件 D 的所有组件）借助于消息 210 请求对象的当前映像，并且起动监控定时器，以用于监控 A 的映像接收。然而，B 事先已经借助于状态转换消息 208（StChID=9，StOwn=B，MID=9）用信令向所有其他组件发送了另一状态转换。

状态转换消息 208 由 C 接收，随后 C 确定，B 丢失了消息，因为 B 的本地 MID 为 20，而接收的消息中包含的是 22。在 MID 与期望值偏离时，不向其他组件发送广播，而是由其 MID 示出误差的那个伙伴组件请求所有状态转换消息标志 StChID 的完整清单（步骤 214），接收所述完整清单（218），并且借助于它确定，需要为哪个对象请求更新的映像。于是，为该对象通过广播要求映像，步骤 220。

在图 2 的情况下，在接收状态转换消息 208 之后，由 A 向具有状态 ID 8 的 C（和可选地也向所有其他组件）传输所请求的对象 X 的映像（步骤 212），其中在完全接收该映像之前，在 C 处已接收了由 D 引起的另一状态转换（步骤 216）。该状态转换 216 与状态转换 208 共同被存储在 C 处，以便能够与所接收的映像匹配。在此，这是必需的，因为该映像是状态 8 的，消息 208 是状态 9 的，以及消息 216 是

状态 10 的。当然，利用消息 214 和 218 已请求和收到 StChID 的所述清单，并且通过广播请求新的对象映像，步骤 220。该广播由 D 应答（步骤 222），因为组件 D 是当前状态占有者。将该对象映像的 StChID 与暂存的状态变化消息及其标志 StChID 进行比较，并且将其 StChID 小于或者等于该映像的 StChID 的所有暂存的状态变化消息舍弃。

在进行该错误消除之后，所有组件 A、B、C、D 再次具有关于对象 X 的统一状态，其特征在于，参数 StChID = 10，StOwn = D 和 MID A: 17，B: 22，C: 35，D: 567。

本发明指出，在通过已识别出无效的本地状态的组件请求对象映像时，如上所述地起动计时，上一次的状态转换消息（借助于该状态转换消息已识别出错误）的状态占有者希望在该计时内得到所述对象映像。如果该对象映像没有被发送，那么具有次低级优先权的组件接管该任务。如果经广播向所有组件发送该对象映像，那么这有利地不需要附加的请求，因为这样具有次低级优先权的组件确定：原来的状态占有者的对象映像未出现。

可选地可以使用 Ping 机制，以便检测状态占有者的故障，并且在定时器运行结束之前已经发送对象的映像到具有无效本地状态的组件。

弱耦合的分布式系统根据本发明进行再同步所利用的最大延迟取决于监控定时器的值，然而，只有在错误情况下才可看出该延迟的干扰，因此现有技术的传输网络（其在进行不可靠的传输时也具有非常小的错误比）能够借助于与本发明有关的简单而不确认的组播消息总共快速地传输消息，其中在监控定时器运行了配置的时间之后，同时获得完全交易可靠的系统的高安全性。

为了执行本发明，只要利用组播消息可以联系上所有组件，就可以使用任意组播机制。如果将例如互联网协议 IP 用作优选的传输协议和将用户数据报协议 UDP 用作广播协议，那么必须例如保证，可寻址所有组件，即如果组件处于不同的 IP 网络中，那么必须例如使用路由器，并对其进行相应地配置。

为了描述不自动支持用于通知状态的组播机制或者自动支持控制任意多个源的设备，可以使用有代表性的对象。该有代表性的对象的特性基本上如同上述对象。然而，为了直接访问该有代表性的设

备，确定在选出的计算机（作为设备的代表）上的、接管与该设备进行实际通信的对象。为了确定该对象，使用上述机制，即根据整体上已知的计算机特性的最小值将所述对象之一本身识别为有代表性的对象，并且利用它的计算机地址设置属性，以便将它通知给其他的对象。如果多个对象应该同时表明是有代表性的对象，那么通过解决冲突如上所述地再次解决该状态。因此保证，对一个设备总是存在一个有代表性的对象，能够如下扩展监控功能，即在确定计算机故障时，所有对象都复位有代表性的计算机的现在的无效属性。此后，根据已知的算法，相应另一有代表性的对象重新接管权限。如果有代表性的对象从显示设备接收（状态）通知，那么该对象相应改变其状态。又自动地向网络的所有计算机上传输该对象。

当然，可以使用其他算法代替示例性所述的确定最小值来选出解决冲突的对象或者来选择有代表性的对象。因此，可以借助于适当的算法对所述激活的对象和其在参与的计算机上的相应代表分配负载。

为了实现本发明，可以使用已知的编程技术特性，以用于简化实现可如此描述的系统。另外，尤其应用反射机制属于简单地实现信息分配、识别冲突和监控基类中的冲突解决，该应用免除了继续实现执行给定的机制的更高层次。

按照本发明的系统尤其适用于以下应用，即其中必须将数据产生器或者产生器（例如传感器）的信息提供给多个数据负载、所谓用户（例如操作员工作席位），而同时不对通信系统造成不必要的附加负荷。这在很多数据产生器和/或产生大量数据的数据产生器的情况下尤其重要。对此一个重要的例子是也具有空间分离的多个操作员工作席位和诸如视频源、接触式传感器、接近式传感器、湿度传感器、烟雾报警器等大量不同数据产生器的监控系统。

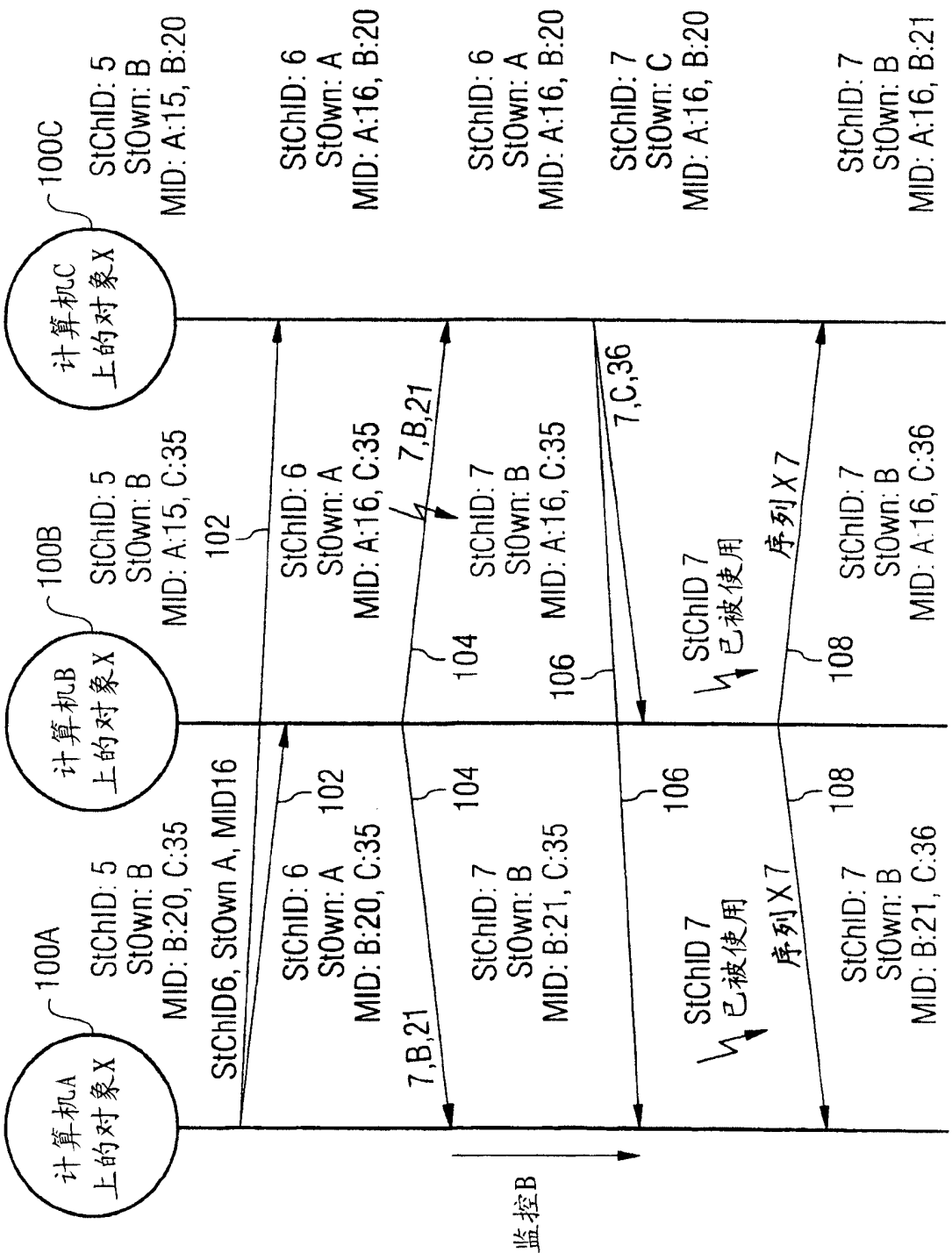

计算机A
上的对象X
100A
StChID: 5
StOwn: B
MID: B:20, C:35
StChID6, StOwn A, MID16
计算机B
上的对象X
100B
StChID: 5
StOwn: B
MID: A:15, C:35
102
计算机C
上的对象X
100C
StChID: 5
StOwn: B
MID: A:15, B:20
StChID: 6
StOwn: A
MID: B:20, C:35
7,B,21
StChID: 6
StOwn: A
MID: A:16, C:35
StChID: 6
StOwn: A
MID: A:16, B:20
104
StChID: 7
StOwn: B
MID: B:21, C:35
StChID: 7
StOwn: B
MID: A:16, C:35
7,C,36
StChID: 7
StOwn: C
MID: A:16, B:20
106
StChID 7
已被使用
序列X7
108
StChID: 7
StOwn: B
MID: B:21, C:36
StChID: 7
StOwn: B
MID: A:16, C:36
StChID: 7
StOwn: B
MID: A:16, B:21
监控B

图 1

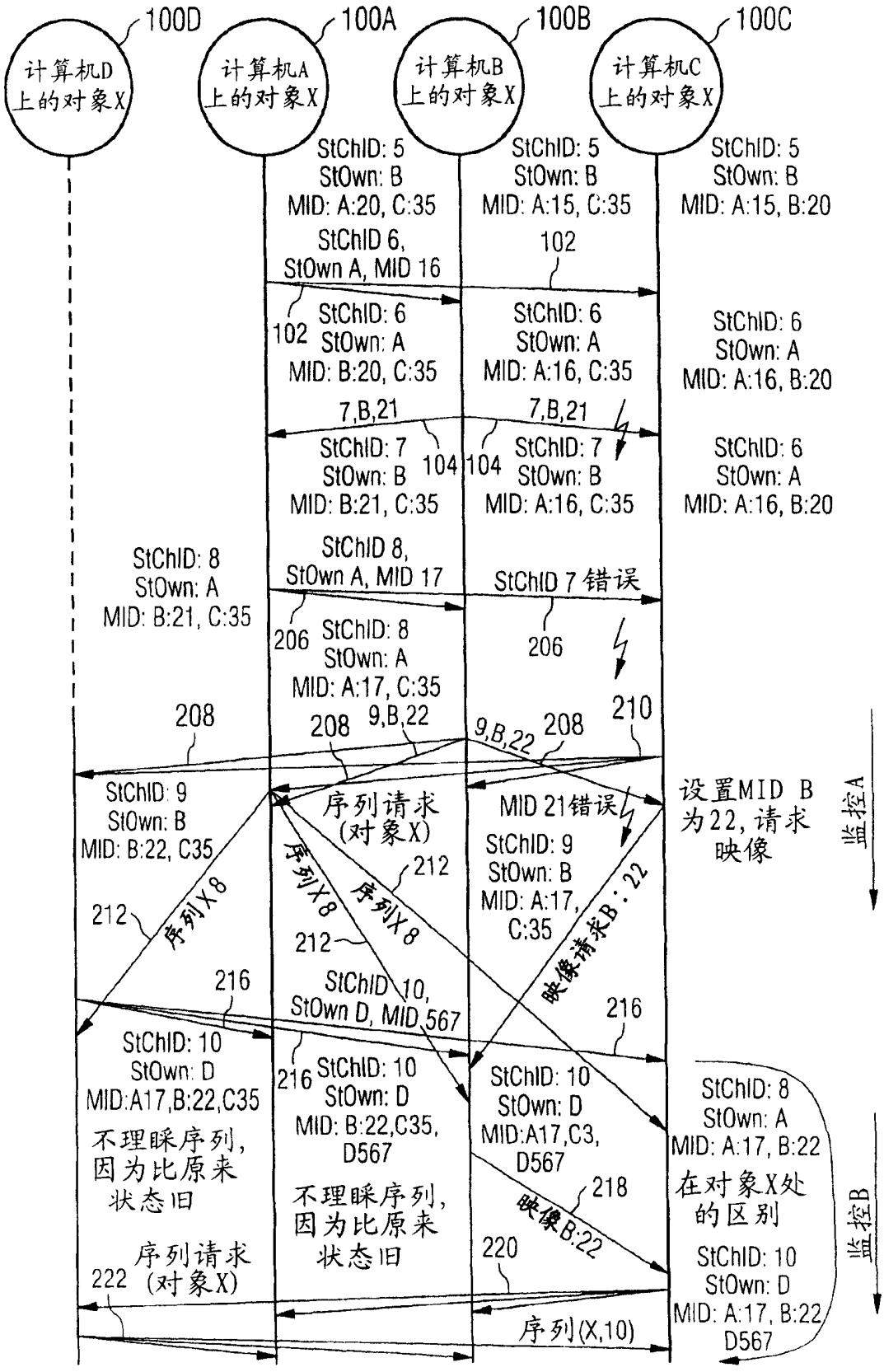

100D
100A
100B
100C
计算机D 上的对象X
计算机A 上的对象X
计算机B 上的对象X
计算机C 上的对象X
StChID: 5 StOwn: B MID: A:20, C:35
StChID: 5 StOwn: B MID: A:15, C:35
StChID: 5 StOwn: B MID: A:15, B:20
StChID 6, StOwn A, MID 16
102
StChID: 6 StOwn: A MID: B:20, C:35
StChID: 6 StOwn: A MID: A:16, C:35
StChID: 6 StOwn: A MID: A:16, B:20
7,B,21
StChID: 7 StOwn: B MID: B:21, C:35
StChID: 7 StOwn: B MID: A:16, C:35
StChID: 6 StOwn: A MID: A:16, B:20
104
StChID: 8 StOwn: A MID: B:21, C:35
StChID 8, StOwn A, MID 17
StChID 7 错误
206
StChID: 8 StOwn: A MID: A:17, C:35
208
9,B,22
210
StChID: 9 StOwn: B MID: B:22, C35
序列请求 (对象X)
MID 21错误
设置MID B 为22,请求 映像
监控A
212
序列X 8
StChID: 9 StOwn: B MID: A:17, C:35
映像请求B:22
216
StChID 10, StOwn D, MID 567
StChID: 10 StOwn: D MID:A17,B:22,C35
StChID: 10 StOwn: D MID: B:22,C35, D567
StChID: 10 StOwn: D MID:A17,C3, D567
StChID: 8 StOwn: A MID: A:17, B:22
不理睬序列, 因为比原来 状态旧
不理睬序列, 因为比原来 状态旧
在对象X处 的区别
218
映像B:22
220
序列请求 (对象X)
222
StChID: 10 StOwn: D MID: A:17, B:22 D567
序列(X,10)
监控B

图 2