[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 17/30 (2006.01)
G06F 9/46 (2006.01)

[12] 发明专利申请公布说明书

[21] 申请号 200810096937.8

[43] 公开日 2008 年 11 月 12 日

[11] 公开号 CN 101303699A

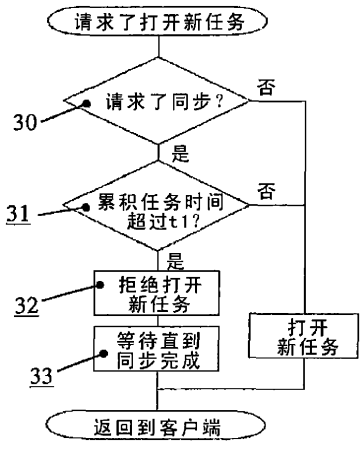
[22] 申请日 2008.5.7
[21] 申请号 200810096937.8
[30] 优先权
[32] 2007. 5. 7 [33] EP [31] 07009132.7
[71] 申请人 软件股份公司
地址 德国达姆施塔特
[72] 发明人 海因兹·库尔登
[74] 专利代理机构 北京东方亿思知识产权代理有限责任公司
代理人 柳春雷
权利要求书 1 页 说明书 6 页 附图 4 页

[54] 发明名称

对访问数据库的多个客户端进行同步的方法和服务器

[57] 摘要

本发明涉及一种对访问数据库(1)的多个客户端(c0 – c5)进行同步方法和服务器，每个客户端(c0 – c5)在数据库(1)上执行多个任务，其中，针对每个客户端(c1 – c5)，所述方法包括以下步骤：累积(41)在发出同步请求之后由客户端(c1 – c5)执行的一个或多个任务的时间；并且如果累积的任务时间超过了最大累积任务时间(t_1)，则拒绝(32)客户端的打开新任务的请求。



1. 一种对访问数据库（1）的多个客户端（c0-c5）进行同步的方法，每个客户端（c0-c5）在所述数据库（1）上执行多个任务，针对每个客户端（c0-c5），所述方法包括以下步骤：

a．累积（41）在发出同步请求之后由所述客户端（c0-c5）执行的一个或多个任务的时间；

b．如果所累积的任务时间超过了最大累积任务时间（t_1），则拒绝（32）所述客户端（c0-c5）的打开新任务的请求。

2．根据权利要求 1 的方法，还包括为所有客户端（c0-c5）定义公共最大累积任务时间（t_1）的步骤。

3．根据权利要求 1 的方法，还包括为每个客户端（c0-c5）单独定义最大累积任务时间（t_1）的步骤。

4．根据前述权利要求中任意一项的方法，其中，如果同步之后执行的任务被打开超过超时时间段（t_0），则终止该任务。

5．根据前述权利要求中任意一项的方法，其中，所述超时时间段（t_0）的值与所述最大累积任务时间（t_1）的值相关。

6．根据前述权利要求中任意一项的方法，其中，当不再有客户端（c0-c5）的打开的任务时，在所述数据库（1）上执行管理任务，特别是所述数据库（1）的备份。

7．根据权利要求 6 的方法，还包括在已执行了所述管理任务之后，发起在步骤 b 中被拒绝的任务的步骤。

8．根据前述权利要求中任意一项的方法，其中，在已执行了所述管理任务之后，对具有被拒绝的任务的所有客户端（c0-c5）执行唤醒。

9．适于执行前述权利要求 1-8 中任意一项权利要求所述方法的数据库服务器（3）。

10．包括用于执行前述权利要求 1-8 中任意一项权利要求所述方法的指令的计算机程序。

对访问数据库的多个客户端进行同步的方法和服务器

技术领域

本发明涉及一种对访问数据库的多个客户端进行同步的方法和服务器，其中，每个客户端在数据库上执行多个任务。

背景技术

数据库是现代信息社会中非常重要的技术工具。通常，许多客户端不断地访问数据库，删除、添加或改变数据库内容。例如在金融机构的数据库的情况下，有持续的事务流程在数据库上执行，这要求对相应数据进行正确技术处理。

出于安全和性能的原因，需要定期地在数据库上执行某些管理任务。一个例子是备份整个数据库系统及其所有数据。然而，可能存在其它的管理任务，例如，将数据库迁移到新的硬件和/或软件系统，或测试其相关的性能。当各个客户端连续访问数据库时，如果要避免所存储数据的不一致性，则可能不执行某些管理任务。例如，如果在从一个账户向另一个转移某些数额期间执行金融机构数据库的备份，因为在其间所发生的备份，所以要避免所保存的数据表现为从第一账户减少了数额但所减少的数额并没有添加到第二账户。这仅仅是一个例子，它显示出针对某些管理任务，需要对所有在数据库上进行操作的客户端进行同步，或至少对会引起这种不一致性的客户端进行同步。

在现有技术中，已知执行如下面参见图 1 和图 2 所描述的这种同步。

图 1 示意性给出了在数据库 1 上操作的多个客户端 c0-c5 的布置。如果客户端 c1-c5 在数据库 1 上都没有任何打开的任务，则客户端 c0-c5 是同步的。在同步请求已被发出后，当目前处于被请求状态（在数据库上没有打开的任务）的客户端想要在数据库 1 上打开新任务时，客户端的请求被拒绝，并且指示客户端等待，直到实现了同步并且完成了管理任务为止

（参考图 2 的流程图）。

如果客户端在同步请求被发出时正在执行某项任务，并且没有在给定的超时时间段内将状态改变到所请求的状态（例如从“打开的任务”到“关闭的任务”），则客户端被重置为所期望的状态，以使其不再妨碍同步和管理任务的随后执行。

然而，在某些情况下，当同步请求被发出时，客户端可以释放与数据库的连接。结果，即使任何其它试图在超时时间段期间打开任务的客户端中的一个或多个客户端的任务可以在超时时间段完全过去之前被执行或完成，所述任何其它试图在超时时间段期间打开任务的客户端也将不得不等待，直到超过该时限。因此，用于处理同步请求的现有技术方法容易对很多的客户端造成大量等待时间，而这仅仅是由于一个客户端失去了与数据库的连接。减小超时时间段是减小其它客户端等待时间的明显方式。然而，减小超时时间段将增加所取消任务的数量，这些所取消的任务必须在实现同步且执行相应的管理任务后重新开始。

考虑到上面，本发明第一技术方案的技术问题在于提供一种将多个客户端进行同步的更高效方法，该方法可减小访问数据库的客户端的等待时间，而不增加所取消任务的数量。

发明内容

现有技术中存在的该问题由根据本申请所述的对访问数据库的多个客户端进行同步的方法来解决。在本发明的一个实施例中，针对每个客户端，所述方法包括以下步骤：

－在发出同步请求后，累积由客户端执行的一个或多个任务的时间；并且

－如果累积的任务时间超过了最大累积任务时间，则拒绝客户端的打开新任务的请求。

由此，不是拒绝任何打开新任务的请求，而是对于这样的客户端允许打开新任务，所述客户端在同步请求被发出之后的所有任务持续时间的总和未超过给定时限。因此，如果存在没有迅速关闭其任务的另一客户端，

例如失去了与数据库的连接的客户端，那么其它客户端可继续工作。结果，将缩短客户端的平均等待时间。

在一个实施例中，所述方法还包括为所有客户端定义公共最大累积任务时间的步骤。在该实施例中，平等地对待所有客户端。另一个极端是这样一个实施例，其中，所述方法还包括为每个客户端单独定义最大累积任务时间的步骤。对最大累积任务时间的单独定义允许通过分配最大累积任务时间的较大值来将某些客户端列为优先，这减小了在处理同步请求期间拒绝这种客户端的新任务的可能性。

很明显还存在这样的实施例，其中，定义客户端的组，并且某个组的所有客户端具有相同的最大累积任务时间，而其中组与组之间的值不同。

在一个实施例中所描述的方法与上述现有技术的方法结合。由此，如果同步后执行的任务被打开超过超时时间段，则终止该任务。而且，超时时间段可以是用于所有客户端的通用参数，或如果将给予某些客户端更多的时间来执行它们的任务，则单独地或按组的方式选择超时时间段。超时时间段的值和最大累积任务时间的值将确定每个客户端的平均等待时间和完成同步请求之前的平均等待时间。在一个实施例中，超时时间段的（一个或多个）值与最大累积任务时间的（一个或多个）值相关。

还可以对所描述方法限定进一步的修改。

根据本发明的另一技术方案，本发明涉及数据库服务器，其适于执行上述方法中的任何一个。根据本发明的又一技术方案，本发明涉及包括用于执行上述方法中任何一个的指令的计算机程序。

附图说明

在下面的详细描述中，参考下列附图来描述当前优选的实施例：

图 1：显示出数据存储器、数据库服务器和多个客户端的示例布置；

图 2：显示出根据现有技术的、在同步请求之后处理客户端的新的打开任务请求的流程图；

图 3a：显示出没有同步请求时正常处理客户端任务的定时图；

图 3b：显示出由根据现有技术的方法所发生的等待时间的定时图；

图 4：显示出根据本发明的一个实施例的、在同步请求后处理客户端的新的打开任务请求的流程图；

图 5：显示出在本发明的一个实施例中处理同步请求的流程图；和

图 6a-6c：显示出累积任务时间和超时时间段的三组不同的值对客户端所发生的所有等待时间的总和和在发出同步请求之后直到实现同步为止的时间的影响。。

具体实施方式

图 1 示意性地给出了包括数据存储器 2 和数据库服务器 3 的数据库 1。数据库服务器 3 利用诸如因特网之类的适当网络技术连接到多个客户端 c0-c5。图 1 的布置仅仅是一个例子，并且对于本发明来说不是必需的。特别地，可以有更多客户端 cn（n>5）连接到数据库服务器 3，为了简单起见在图中没有示出这些客户端。

客户端 c0-c5 在数据库 1 上执行各种任务，如在图 3a 的定时图中示例性示出的。可以看到，任务具有不同的时间长度。此外，当 c0-c5 的某个客户端没有在数据库上执行任何任务时也存在这样的时间段。

图 3a 还示出了在某些任务的处理所花的时间大于某个超时时间段 t_0 时，取消这些任务。通过“X”终止某个任务而示出了该情况。在图 3a 的例子中，超时时间段是 4 分钟。然而，这仅仅是示例值。取决于要执行的任务、整个系统的性能和期望的响应时间，超时时间段可长几个数量级或短几个数量级。

对于某些管理任务，需要同步各个客户端 c0-c5。换言之，需要同步的状态，其中，关闭所有任务，而且其中没有打开新任务，以使得可执行管理任务，例如数据存储器 2 的备份，并且提供一致的数据组。

图 2 示出了现有技术的方法，用于将所有示例客户端 c0-c5 变为同步状态。从图 2 的流程图可以看到，只要客户端试图开始新任务，就在第一步骤 20 中检验是否同时已经请求了同步。如果检验结果为否，则打开新任务（参考图 2 中的右分支）。然而，如果有同步请求，则在步骤 21 中拒绝客户端的“打开新任务”请求。在步骤 22 中，在可打开任务之前，

客户端必须等待，直到实现了同步，而且终止了任何随后的管理任务为止。

图 3b 中示出了对客户端 c0-c5 的任务的定时所产生的效果。可以看到，在 t=0 时发出同步请求。之后，客户端 c0-c5 都不允许开始新任务。客户端 c1-c4 的所期望的新任务移位到同步完成时（例如，执行了备份，这在图中未示出）的时间点。这由水平箭头示出，水平箭头的长度反映了相应任务的延迟。在图 3b 的右侧，示出了客户端的总等待时间是 5.6 分钟。由于客户端 c0，直到真正实现了同步状态的时间稍微多于 3 分钟。在超时时间段 t_0 期满之后，取消 c0 的第一任务。从与图 3a 的比较可以看到，客户端 c3 和 c4 的所请求任务是如此的短，以致于它们可以在等待客户端 c0 的第一任务取消的过程中都完全被执行。

图 4 给出了根据本发明的方法的实施例的示意流程图，其克服了这种低效。在预备步骤（图 4 中未示出）中，如上面所说明的，为客户端 c0-c5 中的一个或多个客户端单独定义最大累积任务时间 t_1，或者为所有客户端 c0-c5 共同定义最大累积任务时间 t_1。当请求新打开的任务时，这里在第一步骤 30 中也检验是否已经发出了同步请求。如果检验结果为是，则不会自动拒绝请求。相反，在步骤 31 中进行比较，以确定在发出同步请求之后由相应客户端执行的任务的累积任务时间是否超过了预定义的最大累积任务时间 t_1。仅在该条件也满足的情况下，在步骤 32 中拒绝所请求的新任务，并且在步骤 33 中，客户端必须等待直到完成同步为止。

图 5 中示出的确定过程，用于确定何时达到同步状态，并确定何时可执行相应的管理任务，例如备份。当关闭任一个客户端的任务时，优选地总是执行图 5 所示的步骤。在步骤 40 中，再次检验是否已请求了同步。在步骤 41 中，如果在达到同步状态之前，相应客户端发出打开新任务的另外请求，则将所处理的任务的时间加到该相应客户端的累积任务时间，以便能与上述的方法步骤 31 进行有意义的比较。在步骤 42 中，检验是否存在具有打开的任务的任何其它客户端。如果检验结果为否，则在步骤 43 中实现同步状态，并且可执行任一管理任务（或多个管理任务）。之后，立即在步骤 44 中向一个或多个客户端发出唤醒呼叫，所述一个或多个客

户端在同步请求被发出之后已请求了打开任务但在上述方法步骤 32 中被拒绝。

图 6a-6c 中示出了所产生的定时图的例子。在这些图中，“X”仍表示超过某超时时间段 t_0 并因此被取消的任务。实心黑点表示这样的情况，其中，客户端的累积任务时间超过了最大累积任务时间 t_1，所以不允许该客户端的新任务。

可以看到，与上述的现有技术的方法相比，使用本发明的实施例，发出同步请求和达到同步状态的时间点之间的平均时间更长。查看图 6a 的图形，其中最大累积任务时间 t_1 被设置为 2 分钟，与图 3b 的图形相比，在更晚的时间点达到同步，即在 9 分钟后达到同步。然而，与图 3b 的图形中的上述 5.6 分钟相比，客户端的总等待时间减小为 1.9 分钟。

通常，总等待时间和最大累积任务时间 t_1 的值是相关的。如果参数 t_1 的值更大，则总等待时间将进一步减小，如比较 t_1 从 2 增加到 6 的图 6a-6c 中的图形所看到的。实际上，所描述的方法使图 6c 中的例子在这样的时间点同步，该时间点不需要客户端 c0-c5 的任何新任务进行等待。如果需要，还可以根据要执行的管理任务的相关性（relevance）来动态地选择参数 t_1。

结果，最大累积任务时间 t_1 的所述参数允许区分客户端任务相对于管理任务的优先级。例如，出于安全的原因，应当定期执行整个系统的备份。然而，相比于以尽可能少的延迟来继续服务各个客户端来说，在所有客户端达到同步以使得能够存储一致的数据组时的精确时间点不是那么相关（relevant），并且可以具有较低的优先级。如上面已经论述的，还可以通过为每个客户端单独选择最大累积任务时间 t_1 和超时时间段 t_0 的值，来改进定时特性。

可由数据库服务器 3 来执行所述方法，该数据库服务器 3 控制数据存储器 2 且服务各个客户端 c0-c5。为此，可以以数据库服务器 3 的硬件和/或软件来实现相应指令。在软件实现的情况下，相应的计算机程序包括执行上述方法步骤的指令。

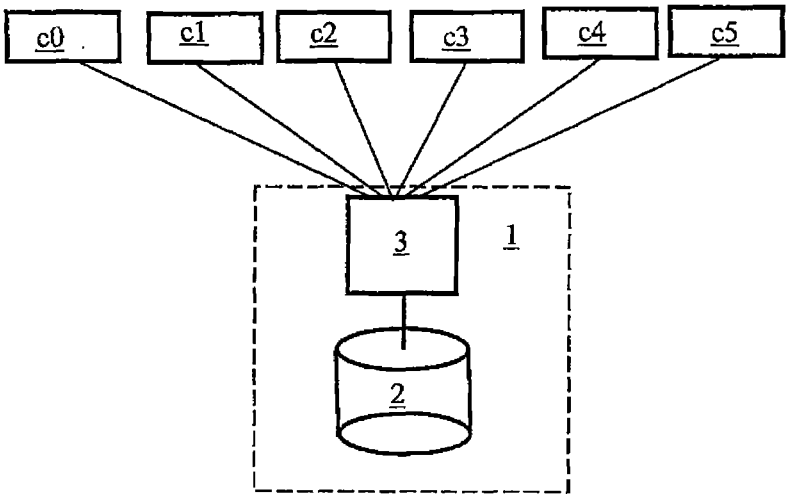

c0
c1
c2
c3
c4
c5
3
1
2

图1

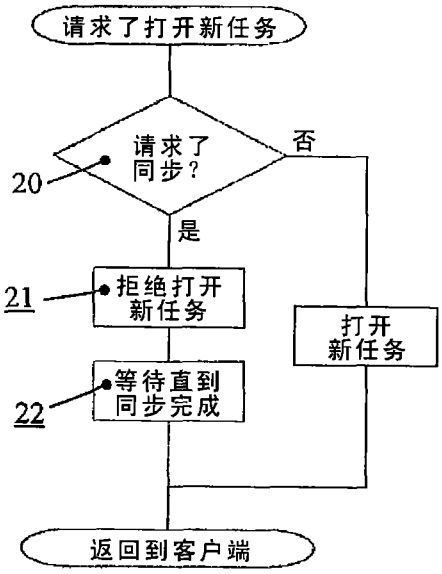

请求了打开新任务
请求了同步？
20
否
是
21
拒绝打开新任务
打开新任务
等待直到同步完成
22
返回到客户端

图2

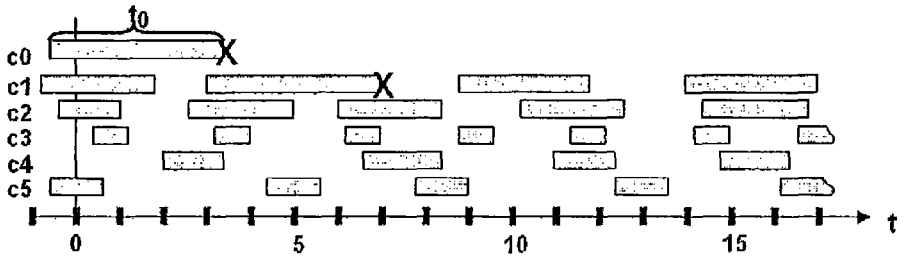

t0
c0
c1
c2
c3
c4
c5
0
5
10
15
t

图3a

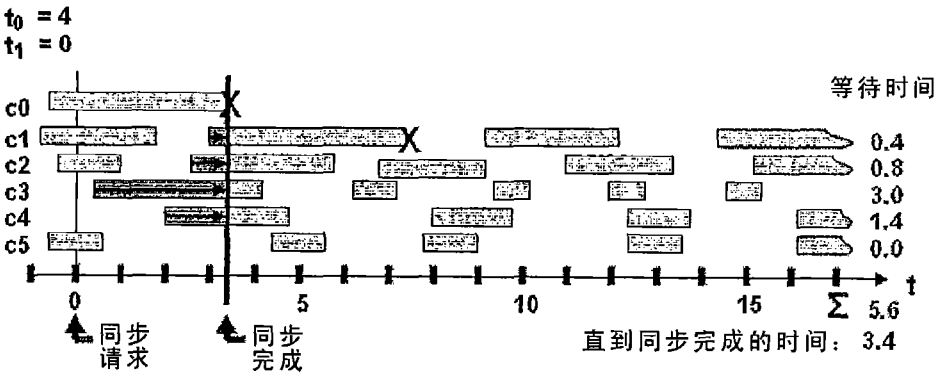

t0 = 4
t1 = 0
等待时间
c0
c1
c2
c3
c4
c5
0.4
0.8
3.0
1.4
0.0
0
5
10
15
Σ 5.6
t
同步请求
同步完成
直到同步完成的时间： 3.4

图3b

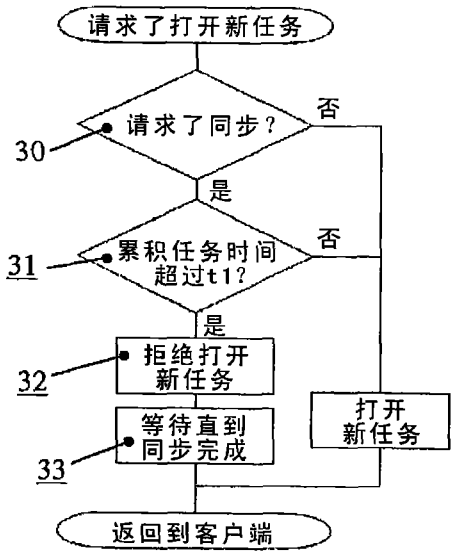

请求了打开新任务
请求了同步？
否
30
是
累积任务时间超过t1?
否
31
是
拒绝打开新任务
32
等待直到同步完成
33
打开新任务
返回到客户端

图4

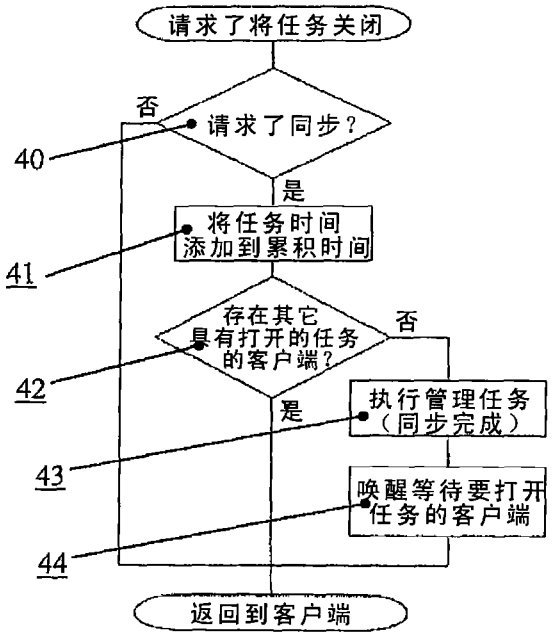

请求了将任务关闭
否
请求了同步？
40
是
将任务时间
添加到累积时间
41
存在其它
具有打开的任务
的客户端？
否
42
是
执行管理任务
（同步完成）
43
唤醒等待要打开
任务的客户端
44
返回到客户端

图5

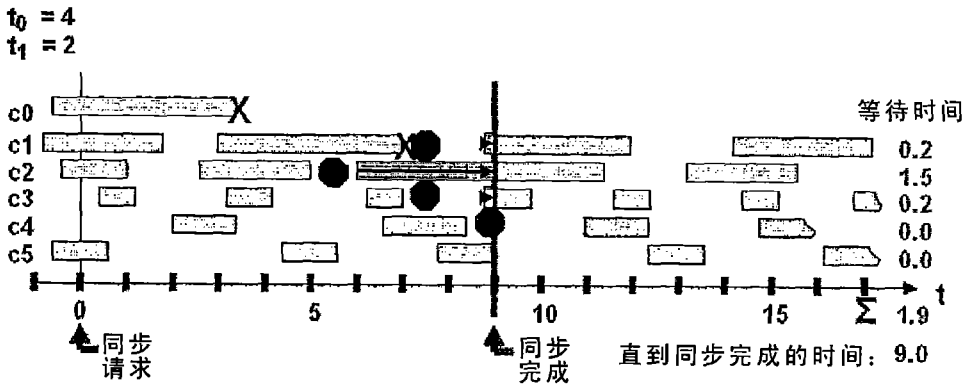

t_0 = 4
t_1 = 2
c0
c1
c2
c3
c4
c5
等待时间
0.2
1.5
0.2
0.0
0.0
0
5
10
15
Σ 1.9
t
同步
请求
同步
完成
直到同步完成的时间：9.0

图6a

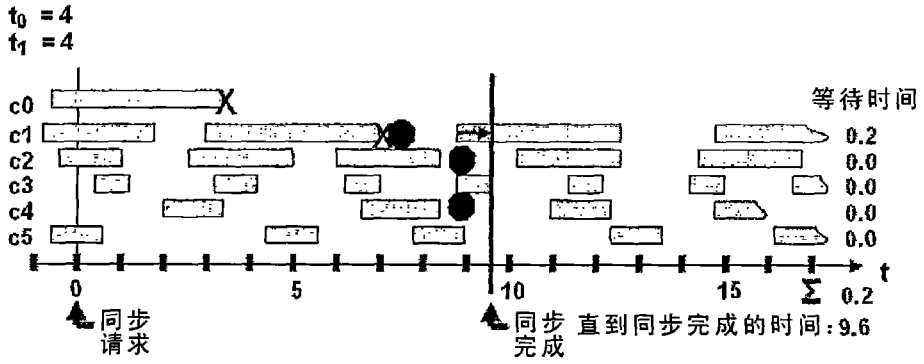

t_0 =4
t_1 =4
c0
c1
c2
c3
c4
c5
等待时间
0.2
0.0
0.0
0.0
0.0
0
5
10
15
Σ 0.2
t
同步请求
同步完成
直到同步完成的时间:9.6

图6b

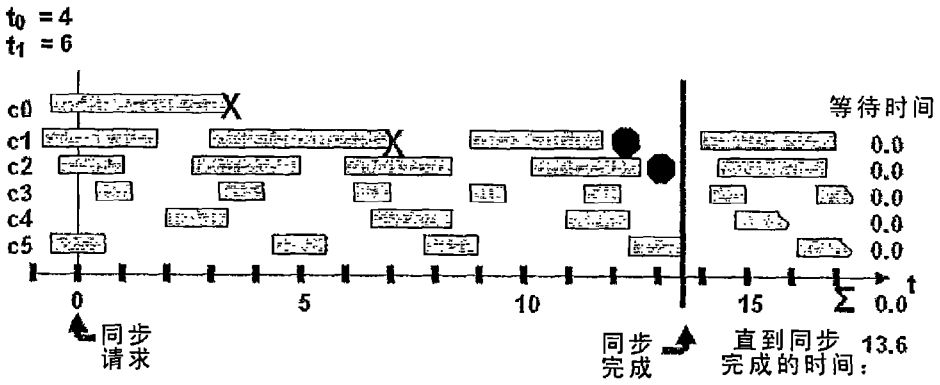

t_0 =4
t_1 =6
c0
c1
c2
c3
c4
c5
等待时间
0.0
0.0
0.0
0.0
0.0
0
5
10
15
Σ 0.0
t
同步请求
同步完成
直到同步完成的时间:
13.6

图6c