(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 1610902 B
(45)授权公告日 2010.05.05

(21)申请号 02826620.X

(22)申请日 2002.11.01

(30)优先权数据
60/330,842 2001.11.01 US
60/365,169 2002.03.19 US

(85)PCT申请进入国家阶段日
2004.07.01

(86)PCT申请的申请数据
PCT/US2002/035083 2002.11.01

(87)PCT申请的公布数据
WO03/038654 EN 2003.05.08

(73)专利权人 弗里塞恩公司
地址 美国加利福尼亚州

(72)发明人 A·N·巴洛 W·F·小哈沃斯
B·T·麦克米伦

(74)专利代理机构 中国专利代理(香港)有限公司 72001
代理人 吴立明 王勇

(51)Int.Cl.
G06F 17/00(2006.01)
G06F 17/30(2006.01)
G06F 12/00(2006.01)

(56)对比文件
US 5995980 A,1999.11.30, 全文.
US 5758150 A,1998.05.26, 全文.
US 5893117 A,1999.04.06, 全文.

审查员 胡徐兵

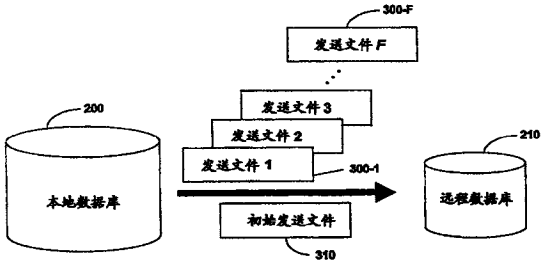
权利要求书 3 页 说明书 16 页 附图 11 页

(54)发明名称

更新远程数据库的方法和系统

(57)摘要

一种通过网络更新远程数据库(210)的方法和系统。基于本地数据库(200)所增加的变化,生成一定数量的的定期更新,称为发送文件(300-F)。每个定期更新包括至少一个事务处理。生成一个初始更新,包括本地数据库在起始时刻的版本,称为初始发送文件。另外,生成一个与起始时刻之前最后的周期更新相关联的标识符,以及一个与起始时刻之前最后所提交的事务处理相关联的标识符。



CN 1610902 B

1. 一种通过网络更新远程数据库的方法，包含：

基于本地数据库所增加的变化来生成多个定期更新，其中，每一个定期更新包括至少一个事务处理；

在生成多个定期更新的同时，通过网络将多个定期更新发送到远程数据库；

生成包括本地数据库在起始时刻的版本的初始更新；

基于所述起始时刻，确定多个定期更新中的最后的定期更新；

基于所述起始时刻，确定最后的事务处理，以及

通过所述网络，将所述初始更新、最后的定期更新标识符以及最后的事务处理标识符发送到远程数据库。

2. 如权利要求 1 所述的方法，其中，发送所述初始更新的步骤包括：

将所述最后的定期更新标识符与所述最后的定期更新相关联；以及

将所述最后的事务处理标识符与所述最后的事务处理相关联。

3. 如权利要求 1 所述的方法，其中所述多个定期更新按规则或不规则的时间间隔来生成。

4. 如权利要求 1 所述的方法，其中，初始更新生成的起始时刻与定期更新生成的起始时刻相同。

5. 如权利要求 1 所述的方法，其中，初始更新生成的起始时刻在定期更新生成的起始时刻之后。

6. 如权利要求 1 所述的方法，其中，定期更新包括多个事务处理，其中，每一个事务处理具有唯一的事务处理标识符。

7. 一种通过网络更新远程数据库的方法，包含：

通过网络接收基于本地数据库所增加的变化的多个定期更新，其中，每一个定期更新包括至少一个事务处理；

通过网络接收初始更新，其包括本地数据库在起始时刻的版本；

从所述初始更新中读取最后的定期更新标识符；

从所述初始更新中读取最后的事务处理标识符；

从最后的定期更新标识符确定最后的定期更新，所述确定最后的定期更新是基于起始时刻进行的；

从最后的事务处理标识符中确定最后的事务处理，所述确定最后的事务处理是基于起始时刻进行的；

将在最后的事务处理之后生成的事务处理应用于远程数据库；以及

将在最后的定期更新之后生成的定期更新应用于远程数据库。

8. 如权利要求 7 所述的方法，进一步包括：

丢弃在初始更新的起始时刻之前生成的定期更新。

9. 如权利要求 7 所述的方法，其中多个定期更新是按周期性的时间间隔而接收的。

10. 如权利要求 7 所述的方法，其中多个定期更新是按周期性的时间间隔成批的接收的。

11. 如权利要求 7 所述的方法，其中的定期更新包括多个事务处理，其中每一个事务处理具有唯一的事务处理标识符。

12. 一种通过网络更新远程数据库的方法，包含：

基于本地数据库所增加的变化而生成多个定期更新，其中每一个定期更新包括至少一个事务处理；

生成包括本地数据库在起始时刻的版本的初始更新、与起始时刻之前最后生成的定期更新相关联的更新标识符，以及与起始时刻之前最后提交的事务处理相关联的事务处理标识符。

13. 如权利要求 12 所述的方法，其中多个定期更新按规则的时间间隔来生成。

14. 如权利要求 12 所述的方法，其中多个定期更新按不规则的时间间隔来生成。

15. 如权利要求 12 所述的方法，其中初始更新生成的起始时刻与定期更新生成的起始时刻相同。

16. 如权利要求 12 所述的方法，其中初始更新生成的起始时刻在定期更新生成的起始时刻之后。

17. 一种通过网络更新远程数据库的系统，包含：

更新生成器，用于基于本地数据库所增加的变化生成多个定期更新，其中每一个定期更新具有至少一个事务处理；

所述更新生成器用于通过网络将所述多个定期更新发送到远程数据库；

耦合至所述网络和所述更新生成器的处理器，用于生成包括本地数据库在起始时刻的版本的初始更新；

所述处理器用于基于所述起始时刻确定多个定期更新中的最后的定期更新；

所述处理器用于基于所述起始时刻确定最后的事物处理；以及

耦合至所述处理器的网络接口，用来通过网络将所述初始更新、最后的定期更新标识符以及最后的事务处理标识符发送到远程数据库。

18. 如权利要求 17 所述的系统，其特征在于，所述处理器用于将所述最后的定期更新标识符与所述最后的定期更新相关联，以及将所述最后的事务处理标识符与所述最后的事务处理相关联。

19. 如权利要求 17 所述的系统，其中多个定期更新按规则或不规则的时间间隔来生成。

20. 如权利要求 17 所述的系统，其中初始更新生成的起始时刻与定期更新生成的起始时刻相同。

21. 如权利要求 17 所述的系统，其中初始更新生成的起始时刻在定期更新生成的起始时刻之后。

22. 如权利要求 17 所述的系统，其中定期更新包括多个事务处理，其中每一个事务处理具有唯一的事务处理标识符。

23. 一种通过网络更新远程数据库的系统，包含：

网络接口，用于通过网络基于本地数据库所增加的变化的来接收多个定期更新，其中每一个定期更新包括至少一个事务处理；

所述网络接口用于通过网络读取包括所述本地数据库在起始时刻的版本的初始更新；

与网络耦合的处理器，用于从所述初始更新中读取最后的定期更新标识符；

所述处理器用于从所述初始更新中读取最后的事物处理标识符；

所述处理器用于基于所述起始时刻从所述最后的定期更新标识符中确定最后的定期更新；

所述处理器用于基于所述起始时刻从所述最后的事物处理标识符中确定最后的事物处理；

与所述处理器通信的远程数据库，用于将在最后事务处理之后生成的事务处理应用于远程数据库，以及

所述的远程处理器用于在最后定期更新之后将生成的定期更新应用于远程数据库。

24. 如权利要求 23 所述的系统，还包括，配置所述系统用来丢弃在初始更新的起始时刻之前生成的定期更新。

25. 如权利要求 23 所述的系统，其中多个定期更新是按规则或不规则的时间间隔来接收的。

26. 如权利要求 23 所述的系统，其中的多个定期更新是按规则或不规则的时间间隔而成批接收的。

27. 如权利要求 23 所述的系统，其中的定期更新包括多个事务处理，其中每一个事务处理具有唯一的事务处理标识符，所述事务处理标识符的顺序是随机的。

28. 一种通过网络更新远程数据库的方法，包含：

基于本地数据库所增加的变化来生成多个定期更新，其中每个定期更新包括至少一个事务处理；

生成初始更新，其包括本地数据库在起始时刻的版本；以及

将所述初始更新与所述定期更新相关联，将更新标识符与在起始时刻之前生成的最后定期更新相关联，以及将事务处理标识符与在起始时刻之前提交的最后事务处理相关联。

29. 如权利要求 28 所述的方法，其中的定期更新包括对域名服务器的更新。

30. 如权利要求 28 所述的方法，其中的更新标识符与对域名服务器的最后定期更新相关联。

31. 如权利要求 28 所述的方法，其中的事务处理标识符与向域名服务器最后提交的事务处理相关联。

32. 一种更新生成器，包含：

基于本地数据库所增加的变化来生成多个定期更新的装置，其中每个定期更新包括至少一个事务处理；

用于通过网络将所述多个定期更新发送到远程数据库的装置；

用于生成包括所述本地数据库在起始时刻的版本的初始更新的装置；

用于基于所述起始时刻确定所述多个定期更新的最后的定期更新的装置；

用于基于所述起始时刻确定最后的事务处理的装置，以及用于通过网络将所述初始更新、最后的定期更新标识符以及最后的事务处理标识符发送到远程数据库的装置。

更新远程数据库的方法和系统

技术领域

[0001] 本发明涉及一种计算机数据库，特别涉及一种可靠的更新数据库的方法和系统。

背景技术

[0002] 随着数据库规模的增加和其结构的高度分布化，要保证网络中相关的数据库所包含的数据版本相同变得越来越困难。如果一个数据库有明显的变化，其它的数据库可能需要尽快更新来产生这些变化。进行上述更新可能导致需要经常在不同的数据库中移动大量的更新信息。这种处理过程的潜在复杂性将是极大的。

[0003] 当系统中的通信不可靠时，这样的问题将进一步扩大。在这种情况下，数据会在传输过程中丢失。于是数据就必须重新传输，并且导致其它的数据库进行再次更新。这样的重复将明显降低系统的工作效率，以及数据库中所包含的最新数据的程度。

发明内容

[0004] 本发明公开了一种通过网络更新远程数据库的方法，由下列步骤构成：

[0005] 基于本地数据库所增加的变化生成多个定期更新，其中每一个定期更新包括至少一个事务处理；

[0006] 在生成多个定期更新的同时，通过网络将多个定期更新发送到远程数据库；

[0007] 生成包括本地数据库在起始时刻的版本的初始更新；

[0008] 基于所述起始时刻，确定多个定期更新中的最后的定期更新；

[0009] 基于所述起始时刻，确定最后的事务处理，以及

[0010] 通过所述网络，将所述初始更新、最后的定期更新标识符以及最后的事务处理标识符发送到远程数据库。

[0011] 本发明还公开了一种通过网络更新远程数据库的方法，由下列步骤构成：

[0012] 通过网络接收基于本地数据库所增加的变化的多个定期更新，其中每一个定期更新包括至少一个事务处理；

[0013] 通过网络接收初始更新，其包括本地数据库在起始时刻的版本；

[0014] 从所述初始更新中读取最后的定期更新标识符；

[0015] 从所述初始更新中读取最后的事务处理标识符；

[0016] 从最后的定期更新标识符确定最后的定期更新，所述确定最后的定期更新是基于起始时刻进行的；

[0017] 从最后的事务处理标识符中确定最后的事务处理，所述确定最后的事务处理是基于起始时刻进行的；

[0018] 将在最后的事务处理之后生成的事务处理应用于远程数据库；以及

[0019] 将在最后的定期更新之后生成的定期更新应用于远程数据库。

[0020] 本发明还公开了一种通过网络更新远程数据库的方法，由下列步骤构成：

[0021] 基于本地数据库所增加的变化而生成多个定期更新，其中每一个定期更新包括至

少一个事务处理；

[0022] 生成包括本地数据库在起始时刻的版本的初始更新、与起始时刻之前最后生成的定期更新相关联的更新标识符，以及与起始时刻之前最后提交的事务处理相关联的事务处理标识符。

[0023] 本发明公开了一种通过网络更新远程数据库的系统，由下列构成：

[0024] 更新生成器，用于基于本地数据库所增加的变化生成多个定期更新，其中每一个定期更新具有至少一个多个事务处理；

[0025] 所述更新生成器用于通过网络将所述多个定期更新发送到远程数据库；

[0026] 耦合至所述网络和所述更新生成器的处理器，用于生成包括本地数据库在起始时刻的版本的初始更新；

[0027] 所述处理器用于基于所述起始时刻确定多个定期更新中的最后的定期更新；

[0028] 所述处理器用于基于所述起始时刻确定最后的事物处理；以及

[0029] 耦合至所述处理器的网络接口，用来通过网络将所述初始更新、最后的定期更新标识符以及最后的事务处理标识符发送到远程数据库。

[0030] 本发明还公开了一种通过网络更新远程数据库的系统，由下列构成：

[0031] 网络接口，用于通过网络接收基于本地数据库所增加的变化的多个定期更新，其中每一个定期更新包括至少一个事务处理；

[0032] 所述网络接口用于通过网络读取包括所述本地数据库在起始时刻的版本的初始更新；

[0033] 与网络耦合的处理器，用于从所述初始更新中读取最后的定期更新标识符；

[0034] 所述处理器用于从所述初始更新中读取最后的事物处理标识符。

[0035] 所述处理器用于基于所述起始时刻从所述最后的定期更新标识符中确定最后的定期更新；

[0036] 所述处理器用于基于所述起始时刻从所述最后的事物处理标识符中确定最后的事物处理；

[0037] 与所述处理器通信的远程数据库，用于将在最后事务处理之后生成的事务处理应用于远程数据库；

[0038] 所述的远程处理器用于在最后定期更新之后将生成的定期更新应用于远程数据库。

[0039] 本发明还公开了一种通过网络更新远程数据库的方法，由下列步骤构成：

[0040] 基于本地数据库所增加的变化生成多个定期更新，其中每个定期更新包括至少一个事务处理；

[0041] 生成初始更新，其包括本地数据库在起始时刻的版本；以及

[0042] 将所述初始更新与所述定期更新相关联，将更新标识符与在起始时刻之前生成的最后定期更新相关联，以及将事务处理标识符与在起始时刻之前提交的最后事务处理相关联。

[0043] 本发明公开了一种更新生成器，由下列构成：

[0044] 基于本地数据库所增加的变化生成多个定期更新的装置，其中每个定期更新包括至少一个事务处理；

[0045] 用于通过网络将所述多个定期更新发送到远程数据库的装置；

[0046] 用于生成包括所述本地数据库在起始时刻的版本的初始更新的装置，

[0047] 用于基于所述起始时刻确定所述多个定期更新的最后的定期更新的装置；

[0048] 用于基于所述起始时刻确定最后的事务处理的装置，以及用于通过网络将所述初始更新、最后的定期更新标识符以及最后的事务处理标识符发送到远程数据库的装置。

附图说明

[0049] 图 1 所示是根据本发明一种实施方式的系统框图。

[0050] 图 2 所示是根据本发明一种实施方式的中心系统框图。

[0051] 图 3 所示是根据本发明一种实施方式，将数据库更新从本地数据库向远程数据库传输的例子。

[0052] 图 4 所示是根据本发明一种实施方式的发送文件。

[0053] 图 5 所示是根据本发明一种实施方式的初始发送文件。

[0054] 图 6 所示是举例说明根据本发明一种实施方式，生成发送文件以及生成初始化发送文件的时间表。

[0055] 图 7 所示是根据本发明一种实施方式，生成本地数据库的更新文件的流程图。

[0056] 图 8 所示是根据本发明一种实施方式，从远程数据库接收来自本地数据库的更新文件的流程图。

[0057] 图 9 所示是根据本发明另一种实施方式，从远程数据库接收并验证来自本地数据库的更新文件的流程图。

[0058] 图 10A 所示是根据本发明一种实施方式的证实更新文件的流程图。

[0059] 图 10B 所示是根据本发明另一种实施方式的证实更新文件的流程图。

[0060] 图 11 所示是举例说明根据本发明一种实施方式的证实更新文件的示意图。

具体实施方式

[0061] 本发明提供一种可靠的通过网络更新远程数据库 210 的方法和系统。在具体实施方式中，基于本地数据库所增加的变化，生成一定数量的的定期更新（以下称为“发送文件”）。每个定期更新包括至少一个事务处理。生成一个初始更新（以下称为“初始发送文件”），包括本地数据库在起始时刻的版本。另外，生成一个与起始时刻之前最后的周期更新相关联的标识符，以及一个与起始时刻之前最后所提交的事务处理相关联的标识符。具体的实施方式还方便地提供对发送文件以及初始化发送文件的分解，来进行远程数据库的可靠的更新。

[0062] 图 1 所示是根据本发明一种实施方式的系统框图。通常，系统 100 具有一个大容量且内置有存储器的数据库，通过网络接收搜索请求并且提供搜索应答。例如，系统 100 可以是一个均衡多处理 (SMP) 计算机，比如，纽约 Armonk IBM 公司制造的 IBM RS/6000® M80 或 S80，加利福尼亚州的 Santa Clara 的 Sun Microsystem，Inc. 制造的 SunEnterpriseTM10000 等等；系统 100 也可以是多处理的个人计算机，比如加利福尼亚州 Palo Alto 的 Hewlett-Packard 公司制造的 CompaqProliantTM ML530（包括两个因特尔奔腾 III866Mhz 处理器）。系统 100 也可以包括一个多处理操作系统，比如IBM AIX®4，

Sun SolarsTM8 操作环境，Red Hat Linux®6.2等等。系统 100 通过网络 124 接收定期更新，同时结合到数据库中。本发明只需通过将每个更新结合到数据库中，而不需要使用数据库锁或者通路控制，就可达到很高的数据库搜索量以及更新量。

[0063] 在一种实施方式中，系统 100 可包括至少一个连接到总线 101 的处理器 102-1。处理器 102-1 可包括一个内部的高速缓冲存储器（例如一个 L1 高速缓存，图中未明示）。一个第二高速缓冲存储器 103-1（例如一个 L2/L3 高速缓存，L2/L3 高速缓存等）位于处理器 102-1 和总线 101 之间。在一种优选的实施方式中，系统 100 可包括一定数量的处理器 102-1……102-P 连接到总线 101。一定数量的第二高速缓冲存储器 103-1……103-P 也可位于一定数量的处理器 102-1……102-P 与总线 101 之间（例如一个浏览结构），或者等同的，至少一个第二高速缓冲存储器 103-1 连接到总线 101（例如一个察看结构）。系统 100 包括存储器 104 连接到总线 101，比如随机存储器 (RAM)，存储一定数量的处理器 102-1……102-P 所执行的信息和指令等。

[0064] 存储器 104 可存储一个大容量数据库，例如，用来传送互联网域名到互联网地址，用来传送名称或电话号码到网络地址，用来提供和更新用户分布的数据，用来提供和更新用户存在的数据等等。有利的，数据库的容量以及每秒传送的数据量都可以很大。例如，存储器 104 可包括至少 64GB 的 RAM，并具有一个 500M 域名记录数据库（即 500×106），一个 500M 用户记录数据库以及一个 450M 记录电话号码的便携式数据库等。

[0065] 在一种 64 比特系统结构的实施方式中，例如，系统包括至少一个 64 比特 big endian 处理器 102-1，连接到至少 64 比特总线 101 和 64 比特存储器 104，使用一个单独无干扰的操作，可将一个 8 字节的指针值写入到存储器地址中的一个 8 字节边界（即存储器地址除以 8 或者例如 8N）。通常，第二高速缓冲存储器 103-1 的存在可简单地延迟 8 字节指针写入存储器 104。例如，在一种实施方式中，第二高速缓冲存储器 103-1 可以是一个使用通写模式的浏览高速缓冲存储器，所以不受到干扰的情况下可使用一个简单的 8 字节存储指令，将 8 字节的数据从处理器 102-1 移动到存储器 104，，并且只需在两个系统时钟回路内。在另一种实施方式中，第二高速缓冲存储器 103-1 是一个使用反写模式的浏览高速缓冲存储器，所以上述 8 字节指针可以首先写入第二高速缓冲存储器 103-1，以后的某一时刻再写入存储器 104，比如，当存储 8 字节指针的高速缓冲存储线写入存储器 104 的时刻（即例如特定的高速缓冲存储线或整个第二高速缓冲存储器“溢出”的时候）。

[0066] 根本的，如果是处理器 102-1，那么当数据被寄存在处理器 102-1 的输出管脚时，所有被第二高速缓冲存储器 103-1 所延迟的 8 字节数据，都以持续无干扰的方式写入存储器 104；如果是处理器 102-2……102-P 的结构，那么一旦数据被寄存在处理器 102-1 的输出管脚上，所有的 8 字节数据都以持续无干扰的方式写入存储器 104，它们被覆盖高速缓冲存储器 103-1……103-P 的内部协议所强制执行，从而延迟了它们写入存储器 104。

[0067] 但是，当一个 8 字节指针值被写入存储器 104 的偏移位置，例如一个跨越 8 字节边界的存储地址，所有 8 字节数据就不能用一个单独的 8 字节存储指令从处理器 102-1 传送。那么处理器 102-1 发出两个单独且不同的指令。例如，如果存储地址开始于 8 字节边界（例如 8N-4）之前的 4 个字节，第一存储指令传送 4 个最重要的字节给存储器 104（例如 8N-4），而第二存储指令传送 4 个最不重要的字节给存储器 104（例如 8N）。重要的是，在这两个单独的存储指令之间，处理器 102-1 可能会被干扰，或者存储器 102-1 可能会将对总线

101 的控制释放给另一系统部件（例如处理器 102-P 等）。因此，位于存储器 104 中的指针值要直到处理器 102-1 完成第二存储指令时才开始有效。如果另一个部件开始向该存储器位置进行单独无干扰的读取，一个无效值将会变回一个可能有效的值。

[0068] 类似的，一个新的 4 字节指针值可利用一个单独无干扰的操作写入除以 4 的存储地址（例如 4N）。注意在以上讨论的实施方式中，一个 4 字节指针值可用一个单独存储指令写入 8N-4 的存储位置。当然，如果一个 4 字节指针值写入一个跨越 4 字节边界的位置，例如 4N-2，那么所有 4 字节数据就不能用一个单独存储指令从 102-1 传送，并且存储器 104 中的指针值可能会无效一段时间。

[0069] 系统 100 也可包括一个只读存储器 (ROM) 106，或者其它静态存储设备连接到总线 101，用于存储处理器 102-1 的静态数据以及指令。一个存储设备 108，例如磁盘或光盘，可连接到总线 101 用于存储信息和指令。系统 100 也可包括显示装置 110（例如 LCD 显示器）和输入装置 112（例如键盘，鼠标，跟踪球等）连接到总线 101。系统 100 也可包括一定数量网络接口 114-1……114-0，可用于接收和发送电、电磁或光信号，它们载有表示不同种类信息的数字信号流。在一种实施方式中，网络接口 114-1 可连接到总线 101 和本地局域网 (LAN) 122，而网络接口 114-0 可连接到总线 101 和广域网 (WAN) 124。一定数量的网络接口 114-1……114-0 可支持不同的网络协议，包括例如吉比特以太网 Gigabit Ethernet（例如 2002 年出版的 IEEE 标准 802.3-2002），FiberChannel（例如 1994 年出版的 ANSI 标准 X.3230-1994），等等。一定数量的网络计算机 102-1……102-N 可连接到 LAN122 和 WAN124。在一种实施方式中，LAN122 和 WAN124 可以是物理上不同的网络，而在另一种实施方式中，LAN122 和 WAN124 也可以经由网关或路由器（图中未不出）。等同的，LAN122 和 WAN124 也可以是相同的网络。

[0070] 如上所述，系统 100 可提供域名服务器 (DNS) 解析服务。在一个 DNS 解析的实施方式中，DNS 解析服务通常在网络传输和数据查找功能中分开。例如，系统 100 可以是一个后端查找引擎 (LUE)，将其优化用于大量数据系列中的数据查找；而一定数量的网络计算机 120-1……120-N 可以是一定数量的前端协议引擎 (PE)，将其优化用于网络处理和传输。LUE 可以是一个在存储器 104 中存储整个 DNS 记录序列的大功率多处理服务器，便于高速，高量的搜索和更新。在另一种等同的实施方式中，DNS 解析服务可由一定数量的大功率多处理服务器或者 LUE 提供，每个服务器或 LUE 都在存储器中存储整个 DNS 记录系列的子集，便于高速，高量的搜索和更新。

[0071] 相反的，所述一定数量的 PE 可以是普通的低分布的基于 PC 的机器，运行一个可有效的多任务操作系统（例如 Red Hat Linux6.2），可最小化 LUE 上网络处理传输负载，从而最大化用于 DNS 解析的可用资源。PE 掌握有线线路 DNS 协议的细微差别，通过 LAN122 应答无效的 DNS 请求并且复用有效的 DNS 请求给 LUE。在另一种等同的实施方式中，具有多个 LUE 存储 DNS 记录子集，PE 将决定哪个 LUE 接收每个有效的 DNS 记录子集，并且复用有效 DNS 请求给合适的 LUE。用于单个 LUT 的 PE 的数目将由，例如，每秒所处理的 DNS 请求的数目和特定系统的性能特征来决定。其它的尺度也可用来决定这个适当的映射比例和工作情况。

[0072] 通常，其它的大量的基于请求的实施方式也可支持，包括例如，电话号码解析，SS7 信号处理，地理位置确定，电话号码到用户的映射，用户位置和其存在的确认等。

[0073] 在一种实施方式中，一个中心在线事务处理(OLTP)服务器140-1可连接到WAN124，并且接收来自不同来源的对数据库142-1所进行的添加，修改和删除（即更新操作）。OLTP服务器140-1可通过WAN124将包括一个本地数据库140-2复制件的更新发送给系统100。OLTP服务器140-1可被优化来以不同的形式和协议进行更新的通信，包括例如超文本传输协议(HTTP)，注册协议(RRP)，可扩展暂时协议(EPP)，服务管理系统/800机制普通接口(MGI)，以及其它的在线暂时协议。一个只读LUE星阵可被应用于一个中心和对话式结构，来提供与其相结合的来自OLTP服务器140-1的高容量、渐增的更新的高速搜索能力。

[0074] 在一种等同的实施方式中，数据可被分散到多个连接到WAN124的OLTP服务器140-1……140-S。OLTP服务器140-1……140-S可从不同来源分别接收对数据库142-1……142-S所进行的添加，修改和删除（即更新通信）。OLTP服务器140-1……140-S可将包括数据库140-1……140-S复制件或者其它动态创建的数据等的更新，通过WAN124向系统100发送。例如，在一种定位的实施方式中，OLTP服务器140-1……140-S可接收远程传感器组的更新通信。在另一种等同的实施方式中，一定数量的网络计算机120-1……120-N也可通过WAN124或LAN122接收添加，修改和删除（即更新操作）。在这种实施方式中，一定数量网络计算机120-1……120-N就像发送请求一样，向系统100发送更新。

[0075] 在DNS解析的实施方式中，每一个PE（例如每一个网络计算机120-1……120-N）可将多个从广域网（例如WAN124）接收的DNS请求报文，结合或者复用成为一个单独的请求复合数据包，并且通过本地局域网（例如LAN122）发送至LUE（例如系统100）。LUE可将多个DNS请求报文应答，结合或者复用成为一个单独的应答复合数据包，并且通过本地局域网发送至适当的PE。通常，应答复合数据包最大长度限定在网络物理层（例如Gigabit Ethernet）的最大传输单元(MTU)之内。例如，典型的DNS请求和应答报文长度分别小于100字节和200字节，这就允许30个请求复用成为一个单独请求复合数据包，15个应答复用成为一个单独的应答复合数据包。但是，一个单独的复合请求数据包中应该包含更小数目的请求（例如20个请求），目的是避免MTU应答时的溢出。如果有更大的MTU长度，请求和应答复用的数目也可相应的增加。

[0076] 每个多任务PE包括一个进入线程和一个输出线程来分别管理DNS请求和应答。例如，进入线程可打乱通过广域网接收的DNS请求数据包的DNS请求组成部分，并且将几个毫秒内的请求复用至一个单独的请求复合数据包。然后进入线程通过本地局域网将请求复合数据包发送到LUE。反之，输出线程可从LUE接收应答复合数据包，将其中的应答解复用，并将不同的字段排列成为一个有效的DNS应答，使其可通过广域网传输。通常，如上所述，其余的大容量的基于请求的实施方式也可是支持的。

[0077] 在一种实施方式中，请求复合数据包也可包括与每个DNS请求相关联的状态信息，例如，来源地址，协议类型等。LUE可将状态信息和相关联的DNS应答包括在应答复合数据包中。然后每个PE利用来自LUE传输的信息建立并回传有效的DNS应答报文。因此，每个PE可方便的作为一个无状态的机器进行操作，即有效的DNS应答可由应答复合数据包中包含的信息形成。通常，LUE可回传应答复合数据包给PE，即发送输入的复合数据包的PE。但是，其它的变形方式明显也可以。

[0078] 在一种等同的实施方式中，每个PE可在请求复合数据包中包括与每个DNS请求相

关联的状态信息，并且包括请求复合数据包的一个状态信息的参数或操作。LUE 可在应答复合数据包中包括状态信息参数和相关 DNS 应答。然后每个 PE 利用来自 LUE 的状态信息参数和随后的状态信息，建立并回传有效的 DNS 应答报文。在这种实施方式中，LUE 可将应答复合数据包回传给 PE，即发送输入的复合数据包的 PE。

[0079] 图 2 所示是根据本发明一种实施方式的中心系统框图。通常，系统可包括一个本地数据库 200（可包含在 OLTP140 中心）和一个或多个远程数据库 210（可包含在 LUE100 中），通过例如互联网或 LAN122 的连接机制连接到本地数据库。数据库可发送和接收更新数据。

[0080] 图 3 是根据本发明的一种实施方式，本地数据库 200 将 F 个发送文件 300-1……300-F 以及初始发送文件 310 发送到远程数据库 210，用于更新远程数据库 210。更新文件可以是单独发送或者成批发送，例如复用发送文件 300，一个发送文件 300 和一个初始发送文件 310，复用发送文件 300 和一个初始发送文件 310，只有发送文件 300，或者只有初始发送文件 310。

[0081] 在本发明的一种实施方式中，处理器 104 可从本地数据库 200 接收包含更新数据的发送文件 300 和 / 或初始发送文件 310。系统 150 可在远程数据库 210 通过通信接口 118 接收发送文件 300 和初始发送文件 310。然后处理器 104 将发送文件 300 或初始发送文件 310 中的更新数据与远程数据库 210 中相应的数据进行比较。如果远程数据库 210 中的数据不同，处理器 104 可将发送文件 300 或初始发送文件 310 应用于远程数据库 210。相应的，远程数据库 210 可依次获得与本地数据库 200 匹配的更新数据。

[0082] 图 4 所示是根据本发明一种实施方式的发送文件 300。发送文件 300 的字段包括，文件标识符 400，文件生成时间 402，文件中的事务处理数目 N404，文件总长度 406，校验位或类似的错误校验指示符 408，以及事务处理 410-1……410-N（包括事务处理标识符）。这些发送文件字段只是举例说明，不能作为本发明实施方式的限制。任何有用的字段都可包括在发送文件 300 中。

[0083] 发送文件 300 包含两个时刻之间本地数据库 200 的变化。这些变化包括例如，新标识符的添加（即数据记录的标识符），已有标识符的删除，与一个标识符相关的一个或多个数据记录的修改，标识符的重新命名，不操作等等。一个或多个这种变化可顺序发生，称为事务处理。发送文件 300 包括这些事务处理的唯一的标识符。这些事务处理可按它们在本地数据库 200 中发生的顺序记录在发送文件 300 中。另外，对于那些包括多个变化的事务处理，这些变化将按它们在本地数据库 200 中发生的顺序记录在事务处理中。

[0084] 通常，事务处理标识符可按任何顺序分配到事务处理中，即事务处理标识符不需要随时间单调递增。例如，依次的两项事务处理的事务处理标识符可首先是 10004 其次是 10002。相应的，事务处理发生的的顺序可由其在当前文件 300-F 或者在先前文件 300-(F-1) 中的位置来决定。通常，事务处理不跨越相邻的文件 300，目的是在一个发送文件的应用过程中完成远程数据库的更新。这可以防止由于网络延时对更新产生的干扰，而这种干扰有可能导致远程数据库 210 中的错误更新。

[0085] 图 5 所示是根据本发明一种实施方式的初始发送文件 310。初始发送文件 310 的字段可包括，例如，文件标识符 500，文件生成时间 502，文件中的事务处理数目 N504，文件的总长度 506，校验位和或类似的错误校验指示符 508，以及整个本地数据库的复制件（数

据)516。初始发送文件 310 还可包括字段 510,表示在文件 310 生成之前所生成的最后一个发送文件 300 的文件标识符 400,以及字段 512,表示在初始发送文件 310 生成之前最后对本地数据库 200 所提交的事务处理。本地和远程数据库 200,210 中的数据可位于数据库 200,210 中的列表中,数据库 200,210 可支持一个任意数目的列表。所以,当数据库包括列表时,初始发送文件 310 可包括一个字段,用于表示每个列表中所记录的记录数目。例如,域名数据库可包括一个域列表和一个名称服务器列表。因此,初始发送文件就包括一个字段,用于表示域名列表中的记录数目,以及一个字段用于表示名称服务器列表中的记录数目。例如列表名称,该字段将指定用于列表中记录索引的密钥,以及列表中的记录数目。另外,初始发送文件 310 可包括一个字段用于表示初始发送文件的版本,通常是 1.0。这些初始发送文件的字段只是举例说明,不能作为本发明实施方式的限制。任何有用的字段都可包括在初始发送文件 310 中。

[0086] 初始发送文件 310 可如前所述包括,例如,整个本地数据库 200 的读出一致的复制件。初始发送文件 310 可在时间点 ts 和 tf 之间与本地数据库 200 形成一致,ts 是初始发送文件 310 生成起始的时刻,tf 是生成完成的时刻。因此,在生成初始发送文件 310 时仅有的操作就是一个“添加”操作。即,在初始发送文件生成时,整个本地数据库 200 在 t 时刻的复制件将记录在初始发送文件 310 中。因此,可执行一个“添加”操作把本地数据库 200 记录到初始发送文件 310。标识符可以以任何顺序记录在初始发送文件 310 中。等同的,如果存在外来的标识符,已参考的数据记录在正参考的数据之前。

[0087] 字段 510 和 512 还可附加提供给初始发送文件 310 一个提示,表示当初始发送 310 生成时,生成发送文件 300 以及向远程数据库 210 提交了。但是,考虑到发送文件 300 和初始发送文件 310 的生成之间缺乏相关性,它们的生成被分解开。这样的结构和过程可防止一种低效率的操作,即在完成初始发送文件的生成之前,发送文件的生成和应用一直被延缓。在本发明的实施方式中,在生成初始发送文件 310 的同时持续生成和应用发送文件 300,对发送文件 300 进行强大的错误校验以及对远程数据库 210 进行系统参数规定,例如,唯一的系统参数或者外来标识符系统参数。对系统规定参数,可阻止那些干扰远程数据库 210 的相关模式的事务处理,从而保护远程数据库 210 中资料的完整。例如,唯一的系统参数可阻止将相同的密钥多次存储于数据库 210 中。

[0088] 图 6 所示是根据本发明一种实施方式的发送文件和初始发送文件的时间表。在图中,发送文件 300(sf-5 到 sf-21) 生成于规则的时间间隔。在一种等同的实施方式中,发送文件也可生成与不规则的时间间隔。通常,发送文件的生成不占用整个的时间间隔。例如,如果发送文件生成的间隔是 5 分钟,它并不占用整个的 5 分钟时间来完成文件的生成过程。另外,如果当生成发送文件 300 的时候,在本地数据库 200 中发生了变化,那么这些变化将由下一个发送文件收集。例如,如果发送文件 sf-4 生成开始于 12:05:00,结束于 12:05:02,那么在 12:05:00 和 12:05:02 之间发生的变化将由收集在 12:05:00 和 12:05:02 之间的变化的发送文件 sf-5(即 300-5) 来收集。

[0089] 发送文件 300-5 和 300-19 在图 6 中说明。这些文件表示,在其它字段之间的文件标识符 601(sf-5,sf-19),文件生成时间 603 以及事务处理标识符 605(例如 10002)。应当注意事务处理标识符可以不是顺序单调递增的。如前所述,事务处理标识符可具有随机值。但是,相关的事务处理自身将以它们在本地数据库中所发生顺序记录于发送文件 300 中。

[0090] 由于初始发送文件 310 的生成和发送文件 300 的生成是分解开的，初始发送文件 310 可生成于任意时刻。例如，初始发送文件 310 可生成于发送文件 300 生成之前，期间或之后。图 6 所示是初始发送文件 310 在第四和第五发送文件（例如 sf-4 和 sf-5）生成之间生成。

[0091] 在一种实施方式中，初始发送文件 310 可在其它字段中包含文件标识符 610(isf-1)，初始发送文件生成之前最后生成的发送文件的文件标识符 615，以及初始发送文件生成之前最后提交的事务处理的事务处理标识符 620。在上述例子中，最后生成的发送文件是发送文件 sf-4，最后提交的事务处理是事务处理 10001。初始发送文件 310 于 12:07:29 开始生成 611。在初始发送文件 310 开始生成时，发送文件 300-5(sf-5) 的前一半，事务处理 10002，100005 以及 10001 都已经向本地数据库 200 提交了。相应的，初始发送文件 310 包括这些事务处理的提示以及将这些事务处理收集于初始发送文件 310 中。但是初始发送文件 310 并不了解在初始发送文件的生成开始之后所依次发生的事务处理 10003 和 10004。

[0092] 当生成初始发送文件 310 时，以发送文件 300-5 开始的发送文件将在规定的间隔生成。这些发送文件可发送到远程数据库 210 并且应用。

[0093] 生成初始发送文件 310 完成于 1:15:29，在第 18 和第 19 个发送文件 300(sf-18 和 sf-19) 的生成之间，不会影响第 19 个发送文件 300-9 的生成、

[0094] 在远程数据库 210 接收到初始发送文件 310 之后，远程数据库 210 将不考虑那些在初始发送文件 310 之前生成的发送文件。因为，初始发送文件 310 已经包括了记录在先前的发送文件中的本地数据库 210 的所有的变化。在上述例子中，远程数据库 210 可不考虑第 1 到第 4 个发送文件 (sf-1 到 sf-4)。发送文件 sf-1 到 sf-4 中所记录的变化已经记录在了初始发送文件 310 中。这些在前发送文件 (sf-1 到 sf-4) 可被删除，替换或存档。类似的，远程数据库 210 可不考虑在初始发送文件 310 生成之前提交的，包含在后来生成的发送文件 300 中的事务处理。初始发送文件 310 在生成时已经包括了这些事务处理。例如，远程数据库不考虑发送文件 sf-5 中的前三项事务处理 10002，10005，10001，因为这些包含在发送文件 sf-5 中的事务处理也记录在了初始发送文件 310 中。这些提交的事务处理可被删除，替换或存档。

[0095] 图 7 所示是根据本发明一种实施例方式的生成本地数据库的更新文件的流程图。系统将基于本地数据库所增加的变化生成 705 一定数量的定期更新，其中每一个更新包括一个或多项事务处理。然后系统把这些定期更新发送 710 到远程数据库。在生成定期更新的同时，系统在起始时刻开始生成 715 一个初始更新。初始更新包括整个本地数据库的版本。系统确定 720 在起始时刻之前生成的最后的一个定期更新以及一个最后提交的事务处理。然后系统发送 725 这些初始更新到远程数据库。初始更新包括一个与最后的定期更新相关联的更新标识符和一个与最后提交的事务处理相关联的事务处理标识符。

[0096] 例如，OLTP140 可在规则或不规则的时间间隔生成 705 发送文件 300。然后 OLTP140 将发送文件 300 发送 710 到远程数据库 210。当生成发送文件 300 时，OLTP140 在起始时刻 611 开始生成 715 初始发送文件 310。初始发送文件 310 可包括整个本地数据库 200 的复制件。然后 OLTP140 确定在初始发送文件 310 生成的起始时刻 611 之前生成的最后一个发送文件 300 以及最后一个提交的事务处理。然后 OLTP140 将初始发送文件发送 725 到远程

数据库 210。初始发送文件 310 包括一个与最后的发送文件 300 相关联的更新标识符 615 和一个与最后提交的事务处理相关的事务处理标识符 620。

[0097] 图 8 所示是根据本发明一种实施方式，远程数据库从本地数据库接收更新文件的流程图。系统接收 805 一定数量的定期更新。每个更新包括一个或多项事务处理。定期更新可单独或成批的接收。系统在某一时刻接收 810 一个初始更新。初始更新包括整个本地数据库的版本。系统从初始更新中读取 815 一个最后定期更新标识符和一个最后事务处理标识符。然后系统确定 820 与更新标识符相关联的最后的定期更新以及与事务处理标识符相关联的最后的事务处理。所述的定期更新和事务处理分别是在初始更新之前最后生成和提交的。系统把相应的定期更新中余下未提交的事务处理应用 825 于远程数据库。然后系统把最后的定期更新之后生成的余下的定期更新应用 830 于远程数据库。应用这些初始更新方便的补充了那些先前丢失的定期更新。

[0098] 例如，LUE100 以规则或不规则的时间间隔接收 805 发送文件 300。发送文件可单独或成批的接收。LUE100 在某一时刻接收 810 一个初始发送文件 310。LUE100 从初始发送文件 310 中读取 815 一个发送文件标识符 615 和一个事务处理标识符 620。然后 LUE100 确定 820 与发送文件标识符 615 相关联的发送文件以及与事务处理标识符 620 相关联的事务处理 605。所述的发送文件和事务处理分别是在初始发送文件 310 生成之前最后生成和提交的。LUE100 把相应的发送文件 300 中余下未提交的事务处理 605 应用 825 于远程数据库 210。然后 LUE100 把最后的发送文件 sf-4 之后余下的发送文件 300 应用 830 于远程数据库 210。

[0099] 在一个等同的实施方式中，例如 LUE100 可丢弃或存档那些没有被应用于远程数据库 210 的，和 / 或那些生成时间 603 在初始发送文件生成时间 611 之前的发送文件 300。丢弃或存档的发送文件 300 包括与发送文件标识符 615 相关联的发送文件 sf-4。

[0100] 应当理解的是，在应用初始发送文件 310 之后，由于远程数据库 210 与初始发送文件 310 形成了读出一致，任何在应用初始发送文件之后应用于远程数据库 210 的发送文件 300 可能丢失。相应的，这些其后的发送文件 300 被再次应用。

[0101] 在本发明的一种实施方式中，发送文件 300 和初始发送文件 310 可从本地数据库 200 传送至远程数据库 210 而不需要应答信号，即不需要 ACK/NACK 信号来指示文件已成功接收。这可大大减少 ACK/NACK 信号所增加的开销。

[0102] 在一种等同的实施方式中，一个 ACK/NACK 信号可从远程数据库 210 发送来指示文件的成功接收。在这种实施方式中，ACK/NACK 信号可以用不可靠的通信在系统中传输。

[0103] 图 9 所示是根据本发明的另一种实施方式，系统验证从本地数据库发送并由远程数据库接收的更新文件的流程图。在此，系统发送 905 一定数量的定期更新。每个更新包括一个或多项事务处理。定期更新可单独或成批发送。然后系统在某一时刻发送 910 一个初始更新并把这个初始更新应用于远程数据库。初始更新包括整个本地数据库的版本。系统首先通过比较数据库来识别 915 本地数据库和远程数据库之间的差异。系统验证 920 这个差异是正确还是错误。然后系统把定期更新按照本发明的一种实施方式应用 925 于远程数据库。这种实施方式可方便的保证在远程数据库中不会接收来自本地数据库的错误更新。

[0104] 例如，OLTP140 以规则或不规则的时间间隔发送 905 发送文件 300 到远程数据库 210。发送文件 300 可单独或成批发送。OLTP140 在某一时刻发送 910 初始发送文件 310 到

LUE100，LUE 可将初始发送文件 310 应用于远程数据库 210。OLTP140 通过比较数据库来识别 915 本地数据库 200 和远程数据库之间的差异。OLTP140 验证 920 这个差异是正确还是错误。然后 OLTP140 通知 LUE100 把发送文件 300 按照本发明的一种实施方式应用 925 于远程数据库 210。然后 LUE 将发送文件 300 应用于远程数据库 210。

[0105] 在一种实施方式中，系统可在标识和验证差异之前应用发送文件和初始发送文件。等同的，系统也可在标识和验证差异之后应用发送文件和初始发送文件。

[0106] 应该理解的是对于那些从来源通过网络传送到目标、目的是应用于目标数据上的任何数据，都应该执行所述验证过程。

[0107] 图 10A 是根据本发明一种实施方式，发送文件和初始发送文件验证的流程图。在将一定数量的定期更新和初始更新发送至远程数据库后，系统将验证这些数据。每个更新包括一个或多个对于本地数据库操作的事务处理。每项事务处理包括一个或多个事件。一个事件是对一个数据库的动作或其发生，例如对于数据库中数据的添加，修改，删除等。

[0108] 首先，系统将远程数据库中的记录和本地数据库中相应的记录进行比较 1000。系统生成 1005 一个描述远程和本地数据库的记录之间的差异的异常信号，其中针对每个差异生成一个异常信号。差异是指相同记录的两个版本之间，至少一个数据值的任何不同。例如，本地数据库中的数据记录是 (12345，xyz.com，123.234.345)，相应的远程数据库中的认为是相同的数据记录是 (12345，abc.com，123.234.345)，可见，在记录的第二数据值就存在一个差异。因此，根据本发明的实施方式就生成一个异常信号来描述这个差异。这个异常信号可以通过简单的指示差异的存在，指示差异的位置，或者描述差异中两个数据值的不同等来描述这个差异。如果认为两个记录包含同样的信息，那么本地数据库中的数据记录就对应于远程数据库中的记录（反之亦然）。

[0109] 应该理解的是所述差异是指存在于一条记录或整个记录中一个或多个数据值的不同。

[0110] 系统可把每个异常信号和异常信号标识符相关联 1010，其中异常信号标识符是和记录的标识符相关联的。例如，数据记录 (12345，xyz.com，123.234.345) 具有标识符 d10。相应的，异常信号标识符也是 d10。每个异常信号可被归类为多个异常信号（或差异）种类中的任意一种。形成一个异常信号列表，包括异常信号的种类以及归类于其中的异常信号标识符。异常信号列表和不同的异常信号种类将在下文进行详细描述。系统也可把事件标识符和每个更新的事件相关联 1015，其中事件标识符是和记录的标识符相关联的。例如，数据记录 (12345，xyz.com，123.234.345) 具有标识符 d10，相应的，事件标识符也可以是 d10。每个更新的事件都可从事件记载表中找到。事件记载表可以是一段时间之内在本地数据库中执行事件的列表等。事件记载表将在下文进行详细的描述。

[0111] 然后系统确定 1020 记录的更新是否有效。图 10B 所示是根据本发明一种实施方式的验证确定的流程图。这种确定将如下所述进行，每个事件与每个异常信号进行比较 1022，如果每个异常信号由事件判断 1024 相符，那么这个更新被指明 1028 是有效的，并且被应用于远程数据库；否则，如果每个异常信号被事件判断 1024 不相符，那么这个更新被指明 1028 是无效的，并且被当作错误存储。当事件标识符对应于异常信号标识符，并且相关的事件对应于和异常信号种类相关联的事件的有效顺序时，那么异常信号判断为相符。所述的有效顺序将在下文进行详细描述。如果异常信号判断为相符，系统将从异常信号列表中除

去异常信号标识符。被判断相符的异常信号是指示这个差异是有效的，比如远程数据库还没有接收到这个更新，且接收到更新后将与本地数据库匹配。

[0112] 在验证期间，系统将在定期和初始更新中识别潜在的错误和故障，保证这些更新在结构和语法上是正确的，使得这些更新能够正确的应用而不产生异常信号或意外的中断，使得在进行本地和远程数据库之间进行比较时可精确的检测到错误，并且使得高分布的数据不会被意外删除。系统将保证定期和初始更新能够成功的应用于远程数据库。

[0113] 通过在验证期间尝试将更新应用于远程数据库，便于一些错误的发现，例如数据中心错误，关于目标已经存在于远程数据库中的警告，或者在尝试应用的过程中发现外来标识符的干扰的警告。因此，在执行完根据本发明一种实施方式所述的验证过程，系统将尝试将这些更新应用于远程数据库中。这种尝试可能失败，即指示在更新中还有其余的错误导致更新无效。相应的，就不会再尝试将这些更新应用于远程数据库了。

[0114] 在一种等同的实施方式中，在执行验证之前，将进行一次尝试，将至少一个更新应用于数据库。如果尝试失败，那么将跳过验证的过程，并且丢弃这个更新。否则，如果尝试成功，那么将执行验证过程，并且保持有效的更新，无效的更新将作为差异存储。

[0115] 在一种实施方式中，OLTP140 将验证发送文件 300 和初始发送文件 310 来保证发送文件 300 和初始发送文件 310 被成功应用于远程数据库 210。

[0116] 在等同的实施方式中，网络计算机 121，LUE100 或者任何已有系统的联合设备也可用于执行验证。

[0117] 关于图 10A，OLTP140 比较本地数据库 200 和远程数据库 210 来确定它们之间的任何异常信号（或者差异）。异常信号包括三种 :数据位于远程数据库 210 中而没有位于本地数据库 200 中 ;数据位于本地数据库 200 中而没有位于远程数据库 210 中，或者相应的数据位于本地数据库 200 以及远程数据库 210 中，但是该数据不同。当然，相应的位于本地数据库 200 以及远程数据库 210 中的数据可以是相同的，这种情况下，数据被认为有效，从而不需要 OLTP140 再进行任何进一步的处理。

[0118] 应该理解的是差异是指一条记录或全部记录中的一个或多个数据值的不同。

[0119] 相应的，OLTP140 比较 1000 本地数据库 200 和远程数据库 210 中的相应记录。OLTP140 生成 1005 一个异常信号，描述位于远程数据库 219 中的记录以及位于本地数据库 200 中的记录的差异，其中对于每个差异生成一个异常信号。OLTP140 把异常信号标识符与每个异常信号相关联，其中异常信号标识符是与记录的标识符关联的。形成一个异常信号列表，包括异常信号的种类，以及归类于其中的异常信号标识符。在一种实施方式中，如果异常信号属于第一种异常信号类型，那么这个异常信号被指定是“列表 1”异常信号（或差异），属于第二种异常信号类型则指定是“列表 2”异常信号。图 11 所示是异常情况列表 1140 举例说明。

[0120] 应当理解的是存在于异常信号列表中的异常信号标识符并不意味着发送文件 300 和初始发送文件 310 不好，因为，所有三种异常信号都是因为本地数据库 200 的变化与应用于远程数据库 210 的更新之间的时延而合理的生成的。例如可能由网络拥堵导致的时延。因此上述验证过程还提供一种从错误数据中除去合理数据的机制。

[0121] 对于初始发送文件 310，OLTP140 通过在数据库 200 和 210 中执行二维全部列表扫描，来比较本地数据库 200 和远程数据库 210。即，本地数据库 200 中的所有的数据将和远

程数据库 210 中的所有数据进行比较。然后，远程数据库 210 中的所有的数据将和本地数据库 200 中的所有数据进行比较。这样更便于提供数据库 200 和 210 之间的无遗漏的比较从而发现所有的差异。

[0122] 对于发送文件 300，OLTP140 只需比较本地数据库 200 和远程数据库 210 中的数据记录。这样更便于提供一种发现目标之间差异的快速请求。

[0123] 等同的，可进行对初始发送文件 310 和发送文件 300 的随机取样。然后 OLTP140 比较本地数据库 200 和远程数据库 210 中的随机取样数据。

[0124] 异常信号列表 1140 可对应于错误的事件，例如，对本地数据库 200 进行的与远程数据库 210 不一致的的添加 (add)，修改 (mod)，和删除 (del)。所以，为了识别这些候选事件，OLTP 140 检查最近向本地数据库 200 提交的事务处理。通常，对应于每个所提交的事务处理，在本地数据库 200 所存储的列表中产生一条目录。这条目录包括改变的记录的标识符，改变的记录的事务处理（或事件）（例如一个添加，修改，和 / 或删除事件），以及一个存储顺序号用于指示事务处理的顺序，等等。

[0125] 一个对应列表 1100 如图 11 所示。在这个例子中，发送文件 300 如列表 1100 所示，包括事务处理 1108-1114。第一目录 1101 指示在第一个事务处理 1108 中，数据（域名服务）n1 和 n2 添加到了与标识符 d1 相关的数据（域）中。因此，标识符是 d1，事件是“添加”，存储顺序号是 11526。类似的，第二目录 1102 指示在第二项事务处理 1109 中，数据 n8 和 n9 添加到了与标识符 d2 相关的数据中。第三目录 1103 指示在第三项事务处理 1110 中，与标识符 d3 相关的数据被删除。第四目录指示在第四项事务处理 1111 中，与标识符 d1 相关的数据被修改并且添加了数据 n5。第五个目录 1105 指示在第五项事务处理 1112 中，数据 n6 和 n7 添加到与标识符 d3 相关的数据中。第六个目录 1106 指示在第六项事务处理 1113 中，与标识符 d4 相关的数据被修改并删除了数据 n3。第 R 个目录指示在第 R 项事务处理 1114 中，与标识符 d5 相关的数据被删除。

[0126] 相应的，如图 10A 所示，OLTP140 把一个事件标识符和更新中的每个事件相关联 1015，其中事件标识符是与记录的标识符相关联的。更新中的每个事件都能从事件记载表中找到。一个根据事件标识符索引化及排序的事件记载表，可从对应列表 1100 产生。一个事件记载表 1120 的例子如图 11 所示。在这，对应列表 1100 中的第一和第四目录 1101，1104 指示与标识符 d1 关联的数据的改变。因此，事件记载表 1120 包括标识符 d1 1121 和两个事件 1126，对与标识符 d1 关联的数据的改变，先是“添加”然后是“修改”。第二目录 1102 指示对与标识符 d2 对应的数据的改变。因此，事件记载表 1120 包括标识符 d2 1122 和一个“添加”事件 1127。事件记载表 1120 包括标识符 d31123 和两个事件 1128，先是“删除”然后是“修改”，由包括对与标识符 d3 关联的第三和第五目录 1103，1105 指示。第六目录 1106 指示对与标识符 d4 关联的数据的改变。相应的，事件记载表包括标识符 d4 1124 和一个“修改”事件 1129。第 R 个目录 1107 指示与标识符 d5 关联的数据的改变，事件记载表 1120 包括标识符 d5 1125 和一个“删除”事件 1130。标识符 1121-1125 按照 d1-d5 的顺序排列。

[0127] 回到图 10A，OLTP140 确定 1020 更新是否有效。这个确定可如下所述执行，例如，根据图 10B 所示的实施方式，首先 OLTP140 比较 1022 事件标识符 1121-1125 和异常信号标识符 1140 来确定哪些标识符是对应的。例如，在图 11 中，事件记载表 1120 中的 d1 事件标

识符 1121 对应于异常信号列表 1140 的“列表 2”中的异常信号标识符。在找到对应的事件和异常信号之后，OLTP140 确定 1024 事件是否判断了这个异常信号。这个判断可如下所述执行。对于事件记载表 1120 中的每个事件标识符 1121-1125，OLTP140 确定事件记载表 1120 中的每个事件 1126-1130 的顺序是否有效。可如下进行，例如通过检查异常情况列表 1140 来确定每个异常情况标识符应属于哪种异常类型，确定对于此种异常情况什么顺序才是一个事件的有效顺序，然后搜索事件记载表 1120 中相应的事件标识符类型以及事件标识符顺序。每个异常类型的有效顺序将在下文进行更详细的描述。如果事件记载表 1120 中事件的一个顺序 1126-1130 与有效的顺序相匹配，那么对应的事件标识符 1121-1125 就是一个有效的顺序。因此，与异常情况标识符关联的异常信号将被判断。并且，包括这些事件标识符的相应的事务处理 1108-1114 也是合法且没有错误的。这种情况下，OLTP140 可从 1140 中除去这些异常情况标识符。

[0128] “列表 1”异常情况类型的有效顺序是（修改）*（删除）。这个顺序可以包括一个先是大于等于零个“修改”事件，然后一个“删除”事件然后是任何事件的顺序。“列表 1”异常情况类型可以是对应于存在于远程数据库 210 中的数据，而不是本地数据库 200 中的数据。在下述情况，数据最近才从本地数据库 200 中删除，这项事务处理还没有写入到发送文件 300。因此，发送文件 300 还没有应用到远程数据库 210。所以数据依然存在于远程数据库 210 中。这可认为是一个合理的差异，因为在这个时刻要生成发送文件 300 并且应用于远程数据库 210。因此，如果在事件记载表 1120 中的异常情况列表 1140 的列表 1 的异常情况标识符中找到一个这样的顺序 1126-1130，相应的事务处理可被认为是有效的。

[0129] 例如在图 11 中，d5 标识符 1125 与其关联的数据从本地数据库 200 中删除，对应列表 1100 中所示，以及在事件记载表 110 中索引化的第 R 个目录 1114。在验证时，d5 被从本地数据库 200 中删除，但是没有从远程数据库 210 中删除。因此，异常情况列表 11140 中包括标识符 d5。根据事件记载表 1120，与 d5 标识符 1125 关联的事件 1130 是“删除”，OLTP140 比较“列表 1”异常情况种类的有效顺序，即（修改）*（删除），与事件记载表 1120 中的 d5 事件 1130。因为“列表 1”的有效顺序和事件 1130 匹配，与标识符 d5 关联的删除事务处理 1114 被认为合理且没有错误。相应的，标识符 d5 从异常情况列表中除去。

[0130] “列表 2”的事件的有效顺序是（添加）。这个顺序可包括一个“添加”操组，然后是任何操作。“列表 2”异常情况类型可对应于存在于本地数据库 200 中的数据，而不是远程数据库 210 的数据。这种情况下，数据最近被添加到本地数据库 200 的事务处理还没有写入发送文件 300。因此，发送文件 300 还没有应用于远程数据库 210。这也可认为是合理的差异因为在这个时刻要生成发送文件 300 并且应用于远程数据库 210。相应的，如果在事件记载表 1120 中的异常情况列表 1140 的列表 2 的异常情况标识符找到一个这样的顺序 1126-1130，相应的事务处理可被认为是有效的。

[0131] 回到图 11，d1 和 d2 标识符 1121，1123 与最初添加到本地数据库 200 中的数据相关联，例如因为它们的事件顺序 1126，1127 以一个“添加”事件开始，d1 和 d2 标识符 1121，1123 与“列表 2”的异常情况类型的有效顺序匹配。相应的包括这些标识符的事务处理 1108，1109 可被认为是有效的，标识符 d1 和 d2 从异常情况列表 1140 中删除。应当注意的是 d3 标识符 1123 在其顺序 1128 中也包括一个“添加”事件。但是，这个“添加”操作不是顺序 1128 中的第一事件。相应的，顺序 1128 不满足“列表 2”的种类。另外，因为 d3 不在

异常情况 1140 的列表 2 中，OLTP140 将不为列表 2 的有效顺序检查它。

[0132] “列表 3”异常情况类型的事件的有效顺序是（删除）（添加）或者（修改）。这个顺序是“删除”事件，其次“添加”事件，其次任何事件或“修改”事件，然后是任何事件。“列表 3”异常情况类型对应于存在于两个数据库 200，210 中的不同的数据。这种情况下，本地数据库 200 的数据最近被修改，这项事务处理没有写入发送文件 300 中。因此，发送文件 300 还没有应用于远程数据库 210。所以，远程数据库 210 中与标识符关联的数据还没有修改。同样的，这也可被认为是合理的差异，因为在相同的时刻，发送文件 300 生成并应用于远程数据库 210。相应的，如果在事件记载表 1120 中的异常情况列表 1140 的列表 2 的异常情况标识符 1126-1130 找到一个这样的顺序，相应的事务处理可被认为是有效的。

[0133] 例如在图 11 中，d3 和 d4 标识符 1123，1124 与在本地数据库 200 中修改的数据关联。在 d3 标识符 1123，d3 标识符 1123 以及其数据最初被删除，并且添加了新的数据的的情况下，事件顺序 1128 包括“删除”随后是“添加”。在 d4 标识符 1124，d4 的资料被修改来除去一些数据，事件顺序 1129 包括“修改”。因为这些事件顺序 1128，1129 与“列表 3”异常情况列表的有效顺序对应，它们的相应事务处理 1110，1112，1113 被认为是有效的，d3 和 d4 标识符从异常情况列表 1140 中删除。

[0134] 关于图 10B，如果在异常情况列表 1140 中的由它们的标识符指示的所有的异常情况已经被事件判断 1024，即如果异常情况列表 1140 是空的，OLTP140 指定 1026 发送文件或初始发送文件 310 有效，并通知 LUE 应用发送文件 300 或初始发送文件 310 于远程数据库 210。然后 LUE100 应用发送文件 300 或初始发送文件 310 于远程数据库 210。

[0135] 相反的，如果所有的异常情况没有被事件所判断 1024，即如果异常情况列表 1140 不是空的，那么余下的异常情况在发送文件 300 或初始发送文件 310 中指示错误。相应的，OLTP140 指定 1028 发送文件或初始发送文件是无效的，并把错误信息存入错误文档。

[0136] 在一种等同的实施方式中，例如，如果发送文件 300 或初始发送文件 310 被指定为无效，在一段预定的时间内，OLTP140 重复验证无效的发送文件 300 或初始发送文件 310，来保证差异的确是错误的。这个预定的时延允许网络有更多的时间来传送一些延迟的发送文件 300，以及允许数据库 200，210 有更多的时间来形成读出一致。

[0137] 在根据本发明的一种实施方式中，远程数据库 210 的数据“滞后”本地数据库 200 的数据一段很明显的时间间隔。相应的，为了比较数据库 200，210 并检测出错误，使数据库 200，210 在同一时刻变为读出一致即它们是互相的精确的复制件，通常远程数据库 210 可运行于本地数据库 200 之前，这样远程数据库 210 中的数据就能与本地数据库中的内容相同。

[0138] 相应的，为了加快验证过程，任何当前生成的初始发送文件 310 和依次生成的发送文件 300 可在确认有效之前应用于远程数据库 210。因此差异的数目就可明显减少。这个把发送文件 300，310 分批的过程可称为打包。信息包中的发送文件 300，310 的第一个和最后一个可分别称为低位和高位水印。第一个信息包，包括初始发送文件 310，称为起始信息包。接下来的所有信息包，称为末端信息包，只包括发送文件 300。

[0139] 打包提供一种成组验证而可以不用单独进行验证。相应的，如果在一个信息包中检测到一个错误，整个信息包将被认为是无效的，而不仅是发生错误的发送文件 300 或初始发送文件 310。

[0140] 根据本发明实施例的结构和方法可用一种常规目的的根据本发明实施例所指教的程序化微处理器来实施。根据本发明的实施方式还包括一个机器可读存媒体，包括用于对处理器进行程序化，来执行根据本发明实施例所述的方法的指令。这种媒体包括而不是仅限定于，任何种类的磁盘，包括软盘，光盘和 CD-ROM。

[0141] 一些本发明的实施方式在此举例说明并描述了。但是，应当理解的是在不脱离本发明的精神和范围的情况下，任何修改和变更都被上述不例所覆盖并且在权利要求书的范围内。

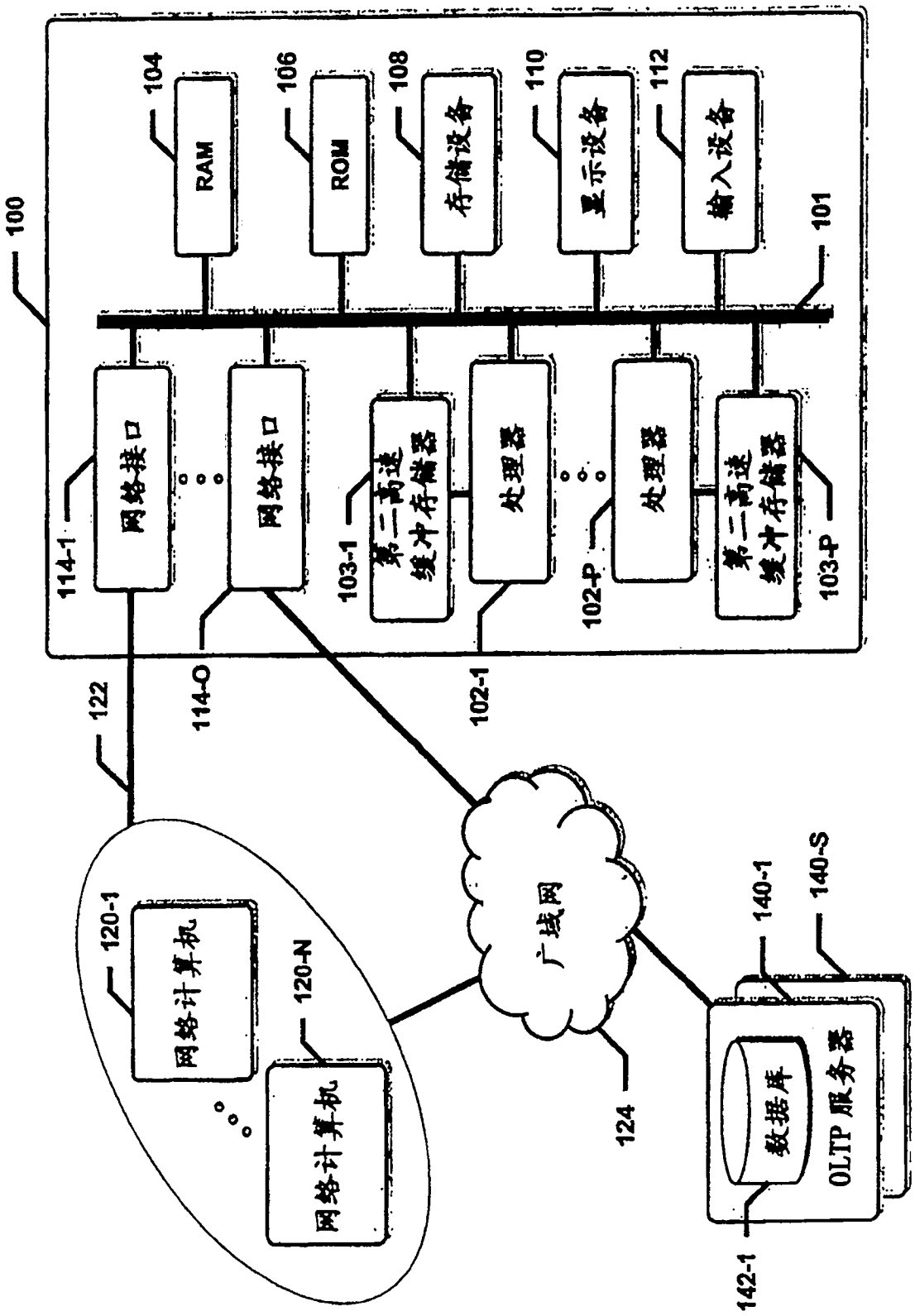

100
104
RAM
106
ROM
108
存储设备
110
显示设备
112
输入设备
101
网络接口
114-1
网络接口
114-O
第二高速缓冲存储器
103-1
处理器
102-1
处理器
102-P
第二高速缓冲存储器
103-P
122
广域网
124
120-1
网络计算机
120-N
网络计算机
140-1
140-S
数据库
OLTP 服务器
142-1

图 1

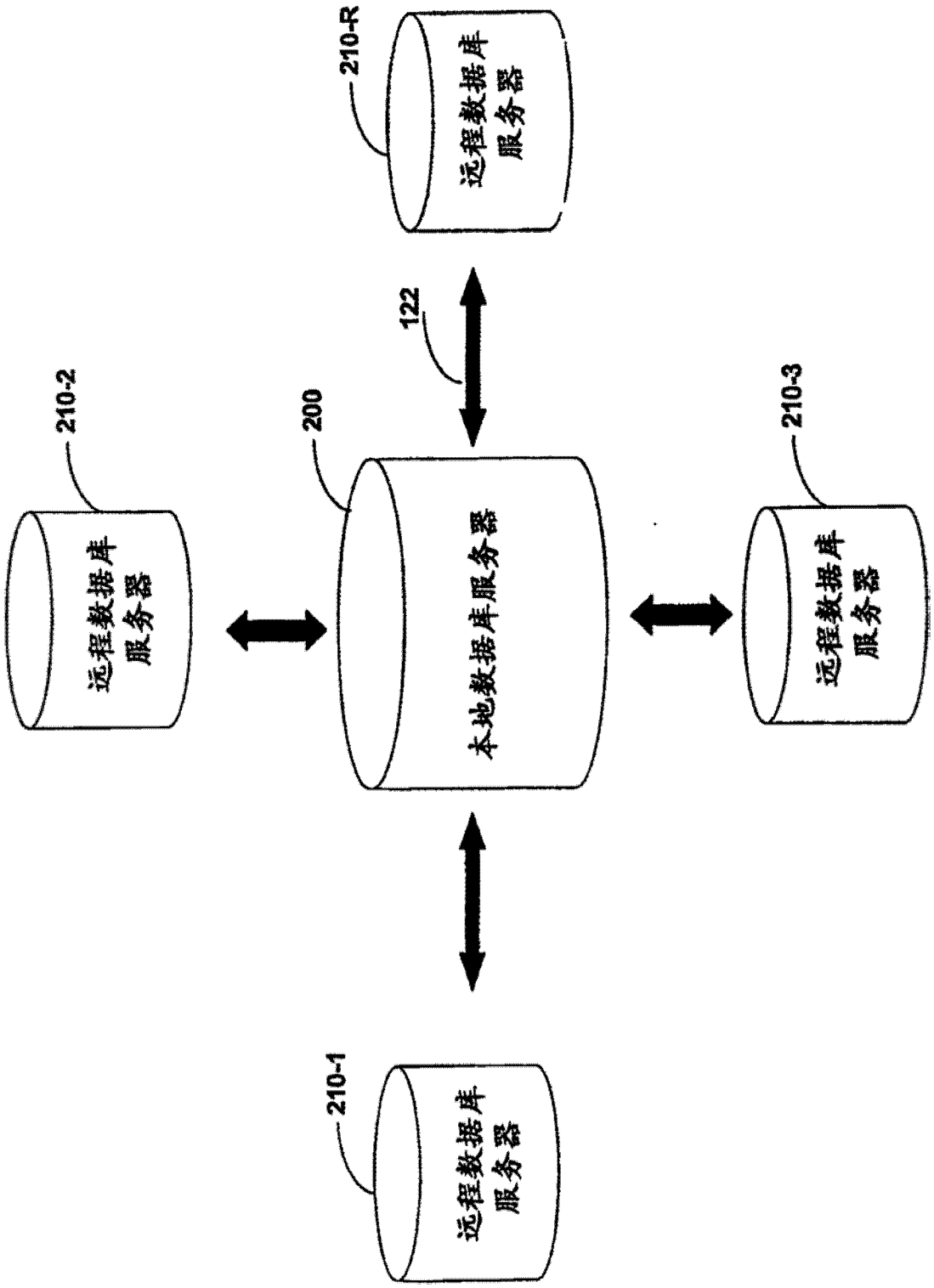

210-R
远程数据库服务器
122
210-2
200
210-3
远程数据库服务器
本地数据库服务器
远程数据库服务器
210-1
远程数据库服务器

图 2

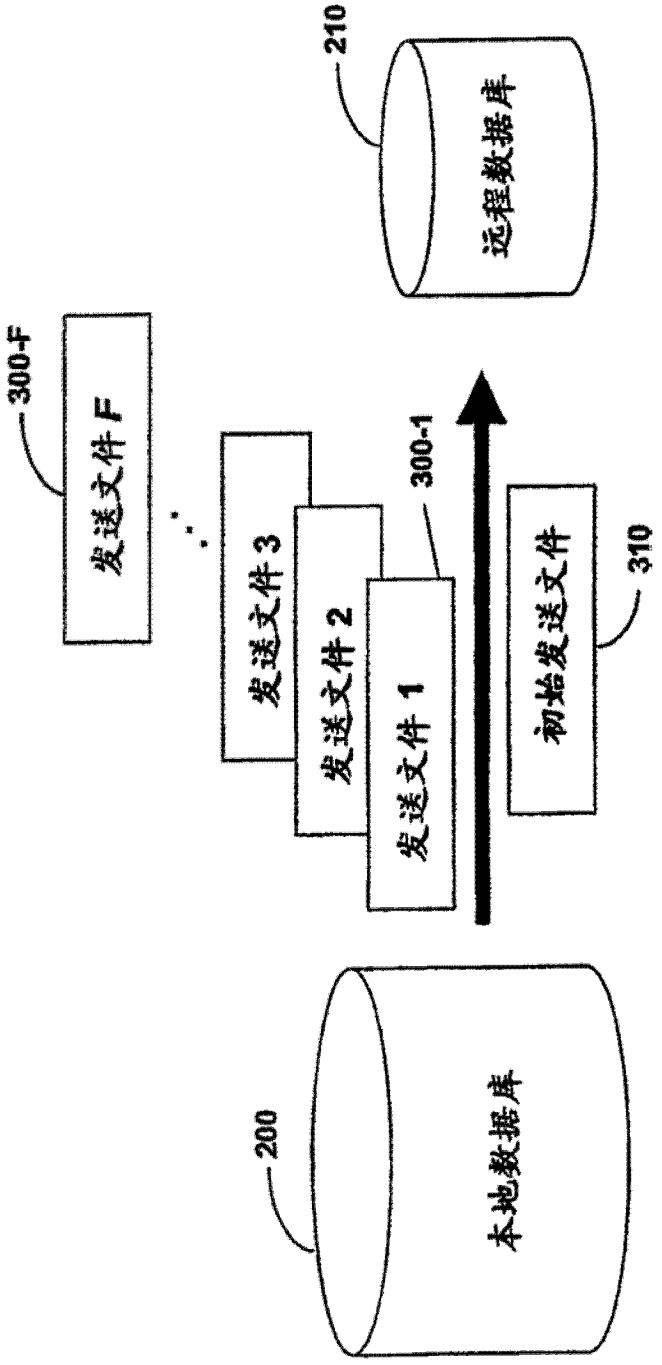

210
远程数据库
300-F
发送文件 F
发送文件 3
发送文件 2
发送文件 1
300-1
初始发送文件
310
200
本地数据库

图 3

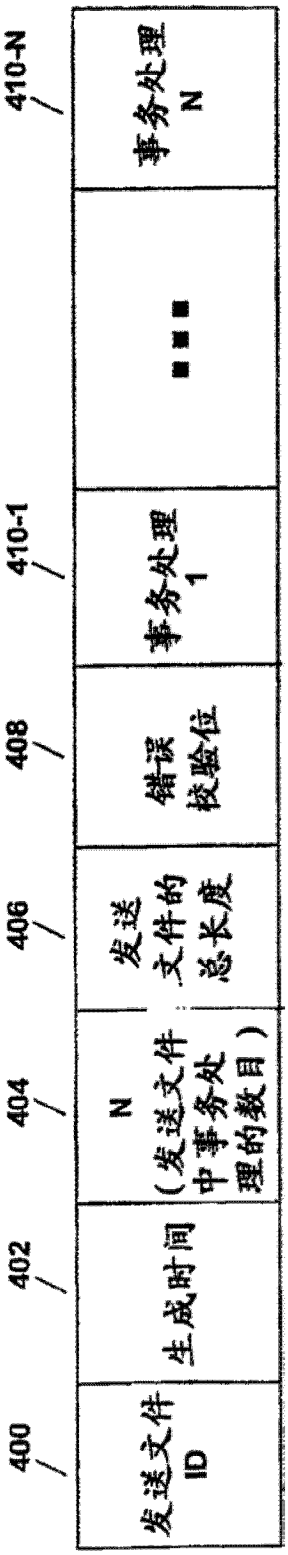

300
400
发送文件 ID
402
生成时间
404
N (发送文件中事务处理的数目)
406
发送文件的总长度
408
错误校验位
410-1
事务处理 1
■■■
410-N
事务处理 N

图 4

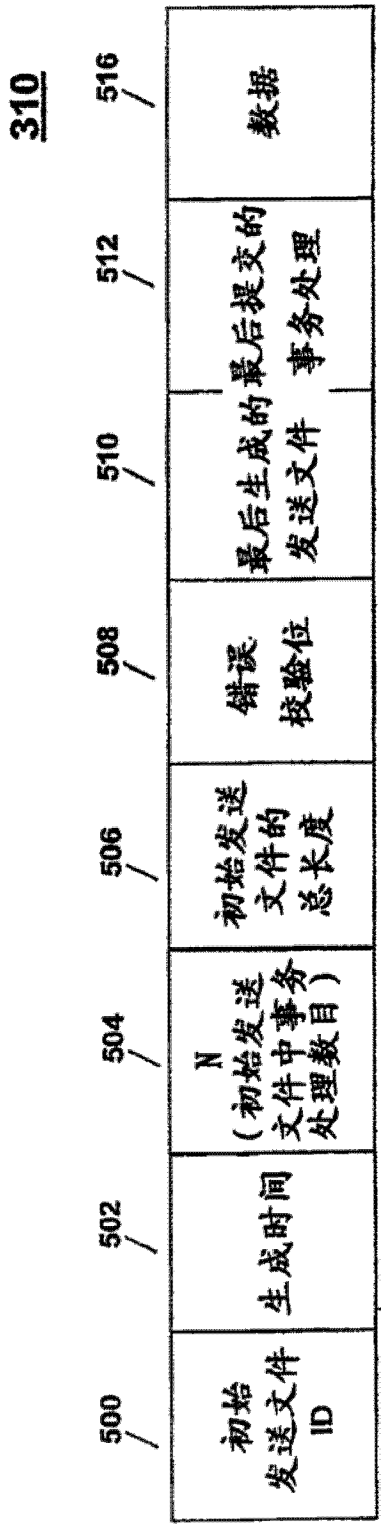

310
500
初始发送文件ID
502
生成时间
504
N（初始发送文件中事务处理数目）
506
初始发送文件的总长度
508
错误校验位
510
最后生成的发送文件
512
最后提交的事务处理
516
数据

图 5

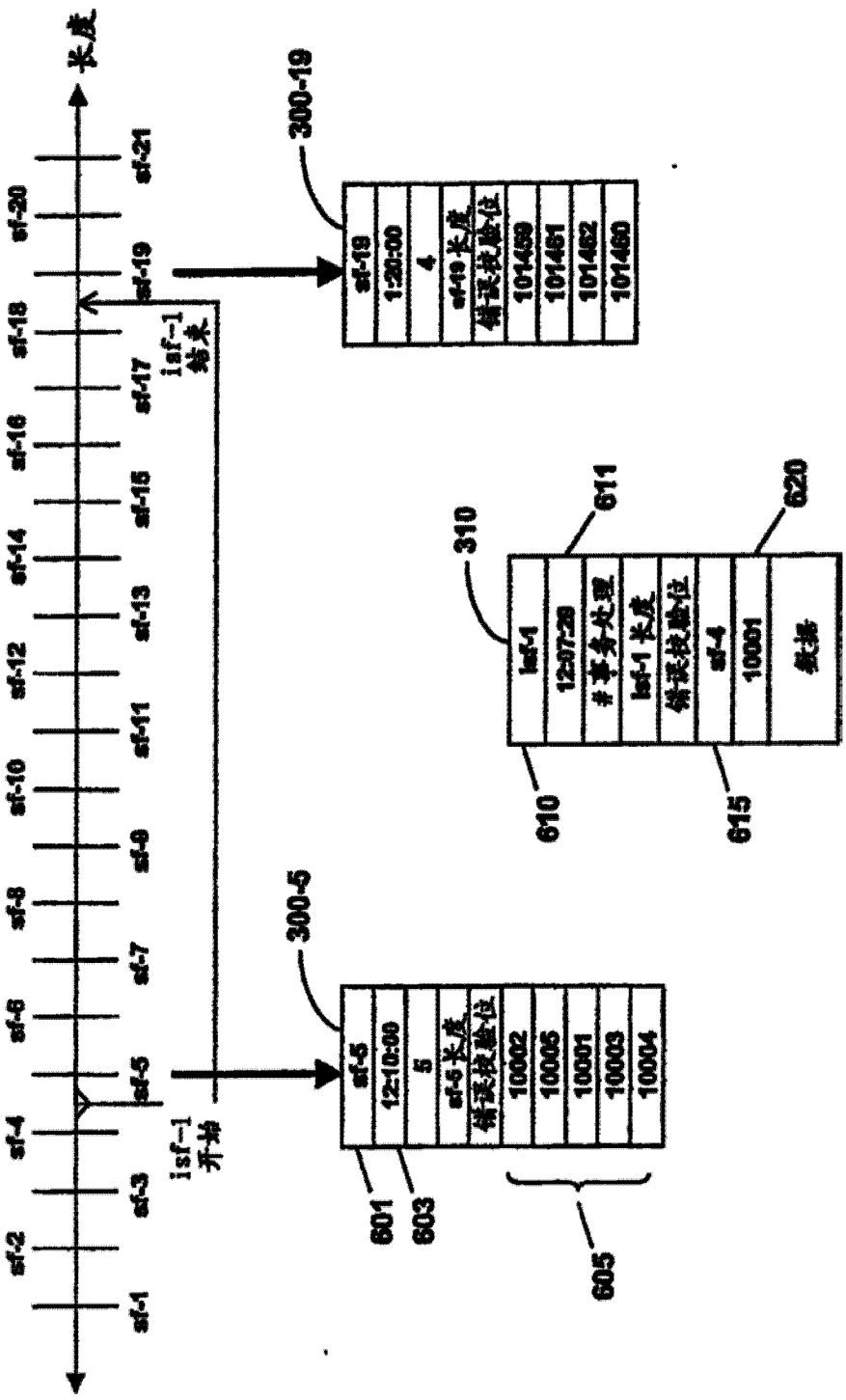

长度
sf-1
sf-2
sf-3
sf-4
sf-5
sf-6
sf-7
sf-8
sf-9
sf-10
sf-11
sf-12
sf-13
sf-14
sf-15
sf-16
sf-17
sf-18
sf-19
sf-20
sf-21
isf-1 开始
isf-1 结束
300-19
sf-19
1:20:00
4
sf-19 长度
循环块单位
101459
101461
101462
101460
310
611
620
isf-1
12:07:28
#事务处理
isf-1 长度
循环块单位
sf-4
10001
截止
610
615
300-5
sf-5
12:10:00
5
sf-5 长度
循环块单位
10002
10005
10001
10003
10004
601
603
605

图 6

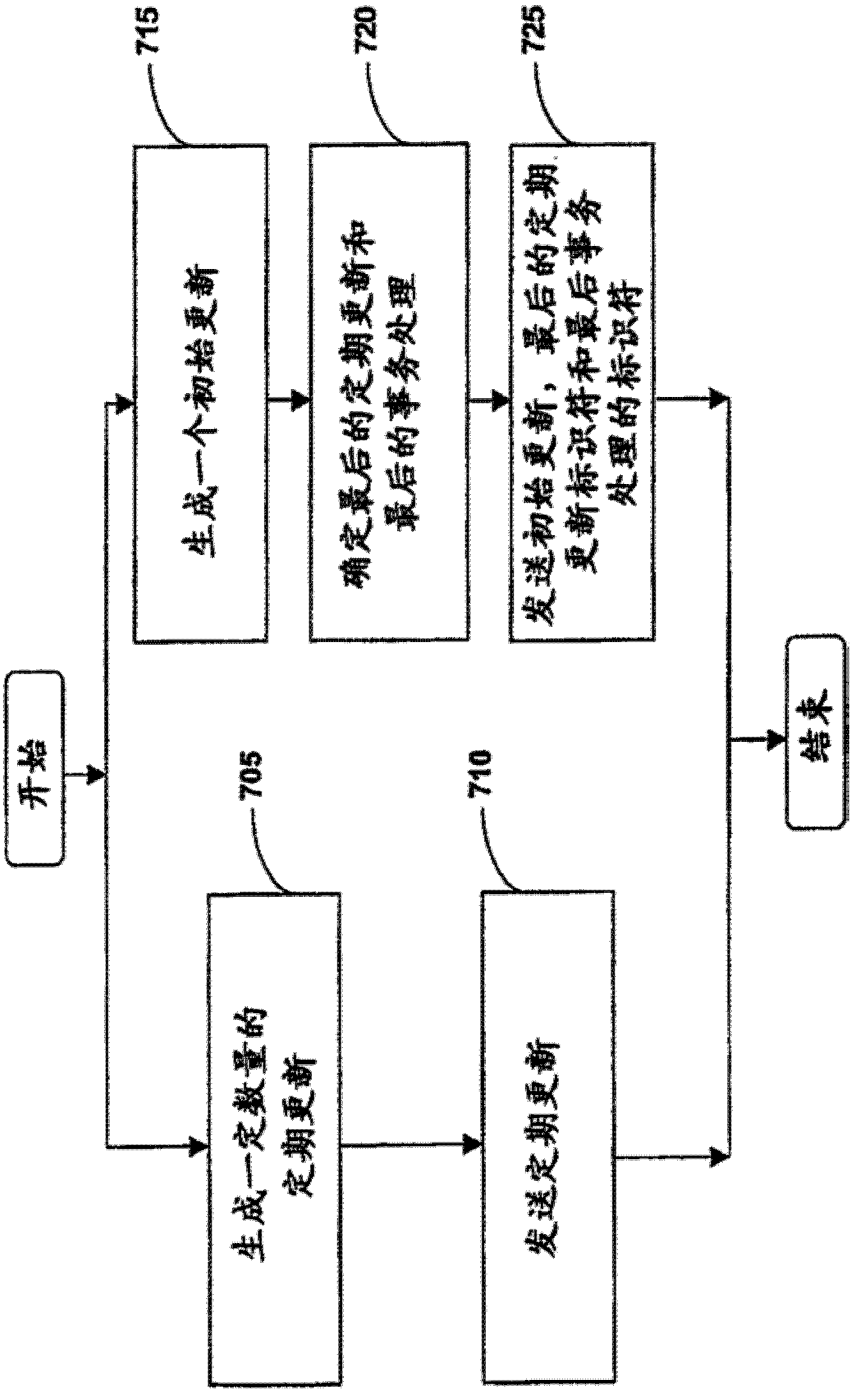

715
720
725
生成一个初始更新
确定最后的定期更新和最后的事务处理
发送初始更新，最后的定期更新标识符和最后事务处理的标识符
开始
结束
705
710
生成一定数量的定期更新
发送定期更新

图 7

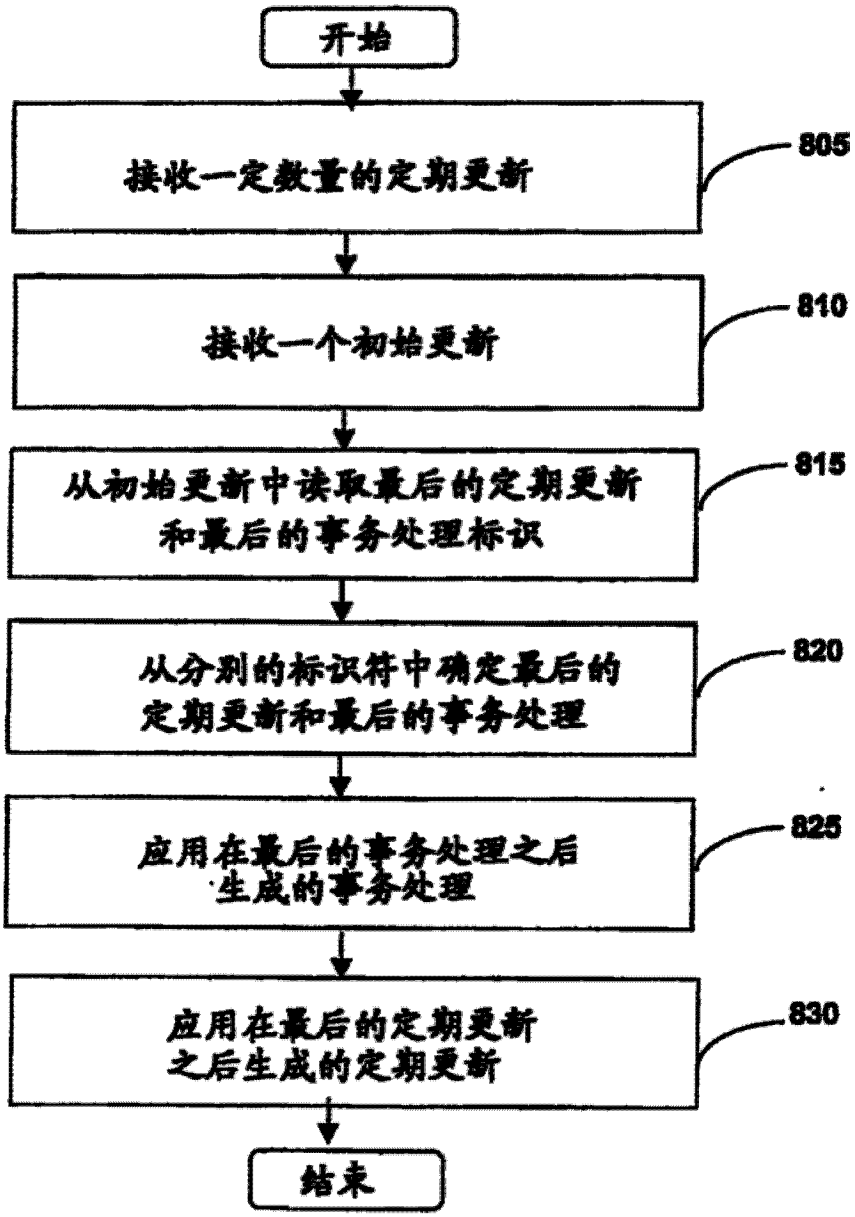

开始
接收一定数量的定期更新
805
接收一个初始更新
810
从初始更新中读取最后的定期更新和最后的事务处理标识
815
从分别的标识符中确定最后的定期更新和最后的事务处理
820
应用在最后的事务处理之后生成的事务处理
825
应用在最后的定期更新之后生成的定期更新
830
结束

图 8

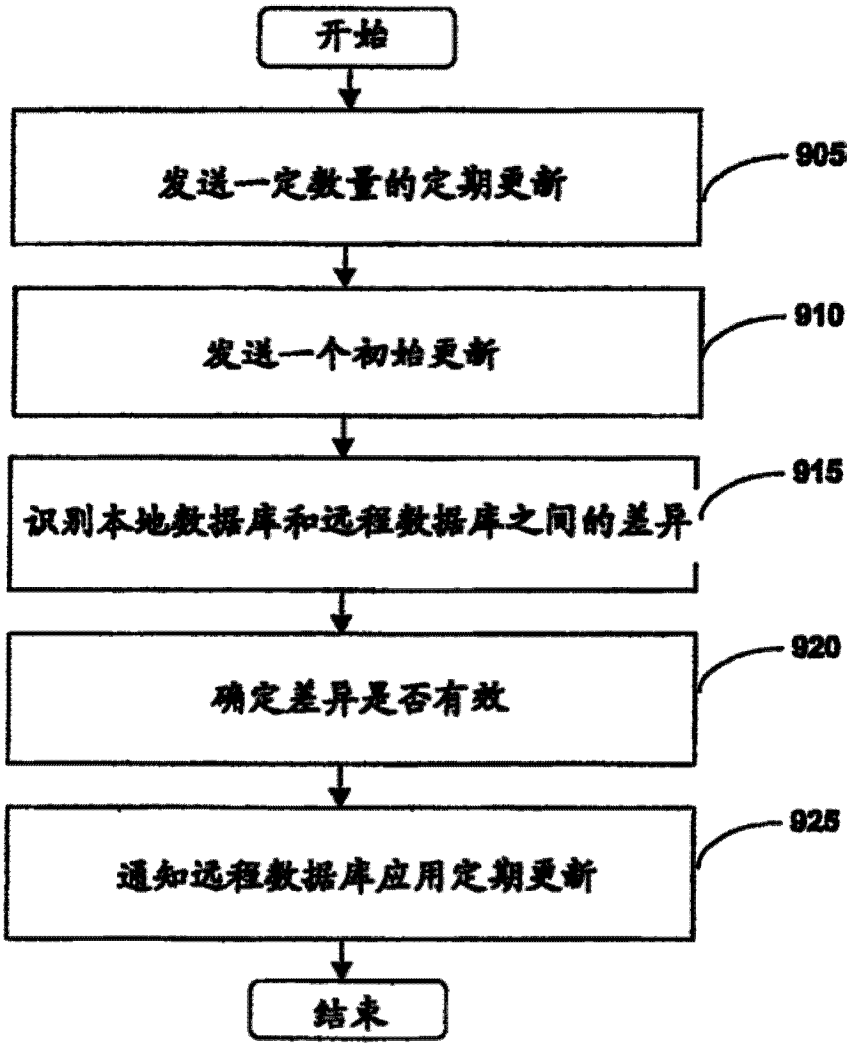

开始
发送一定数量的定期更新
905
发送一个初始更新
910
识别本地数据库和远程数据库之间的差异
915
确定差异是否有效
920
通知远程数据库应用定期更新
925
结束

图 9

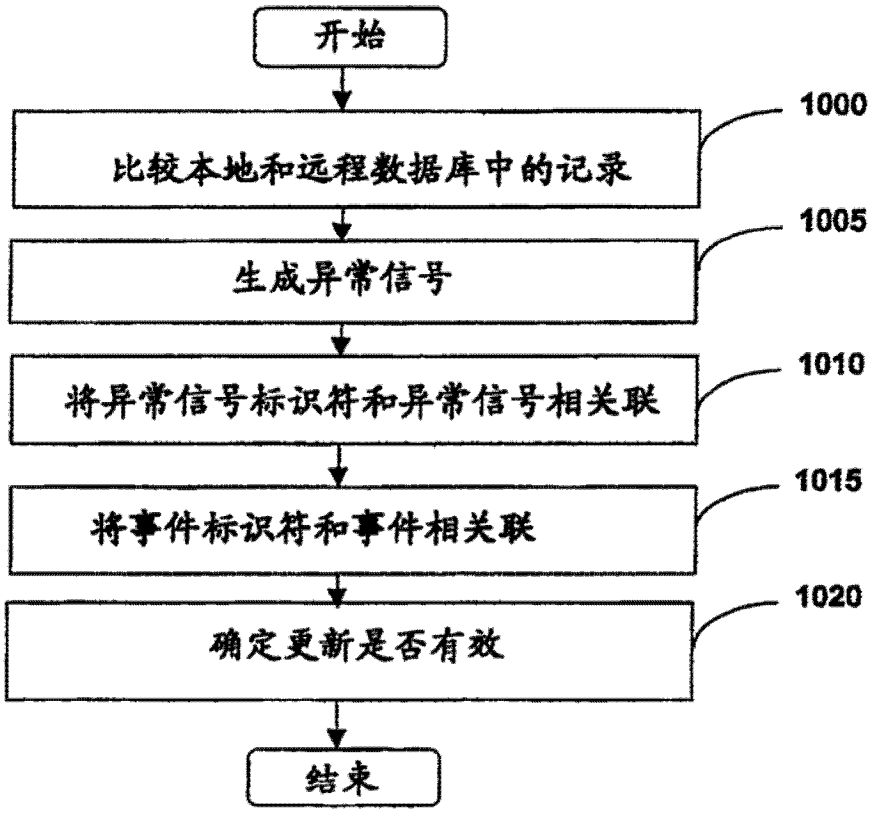

开始
比较本地和远程数据库中的记录
1000
生成异常信号
1005
将异常信号标识符和异常信号相关联
1010
将事件标识符和事件相关联
1015
确定更新是否有效
1020
结束

图 10A

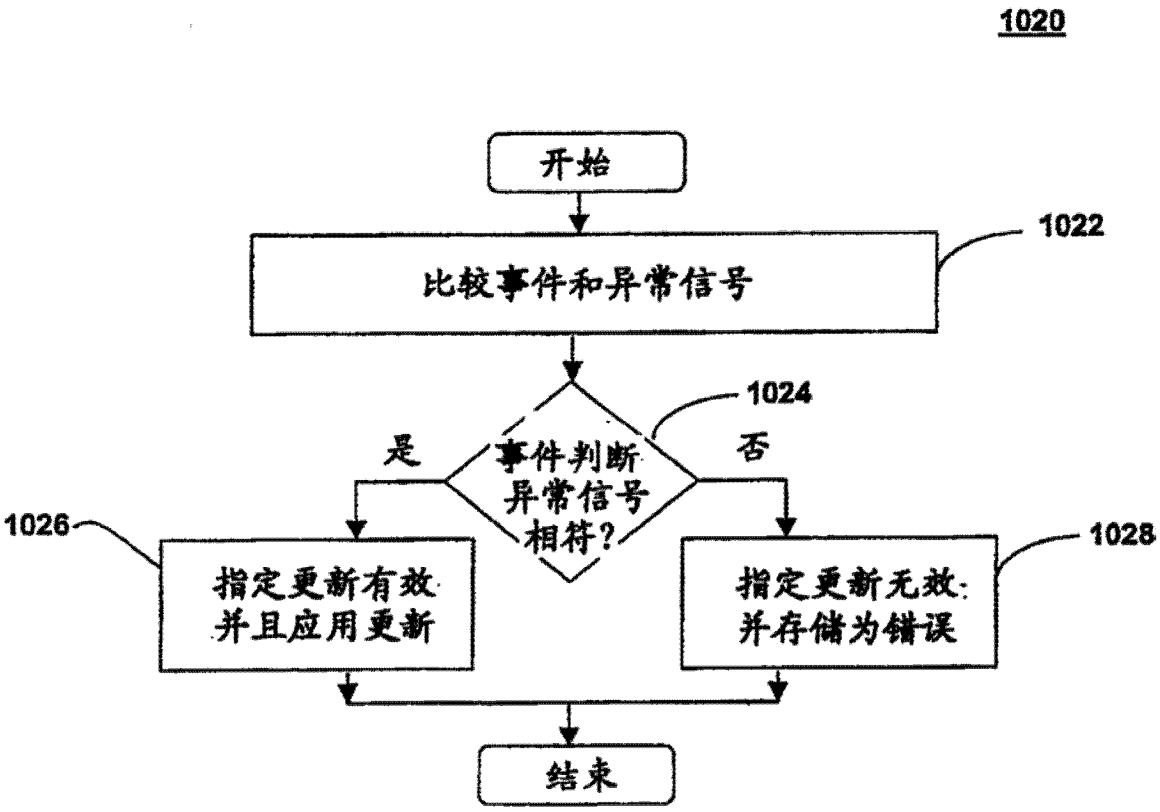

1020
开始
比较事件和异常信号
1022
事件判断异常信号相符?
1024
是
否
1026
指定更新有效并且应用更新
指定更新无效并存储为错误
1028
结束

图 10B

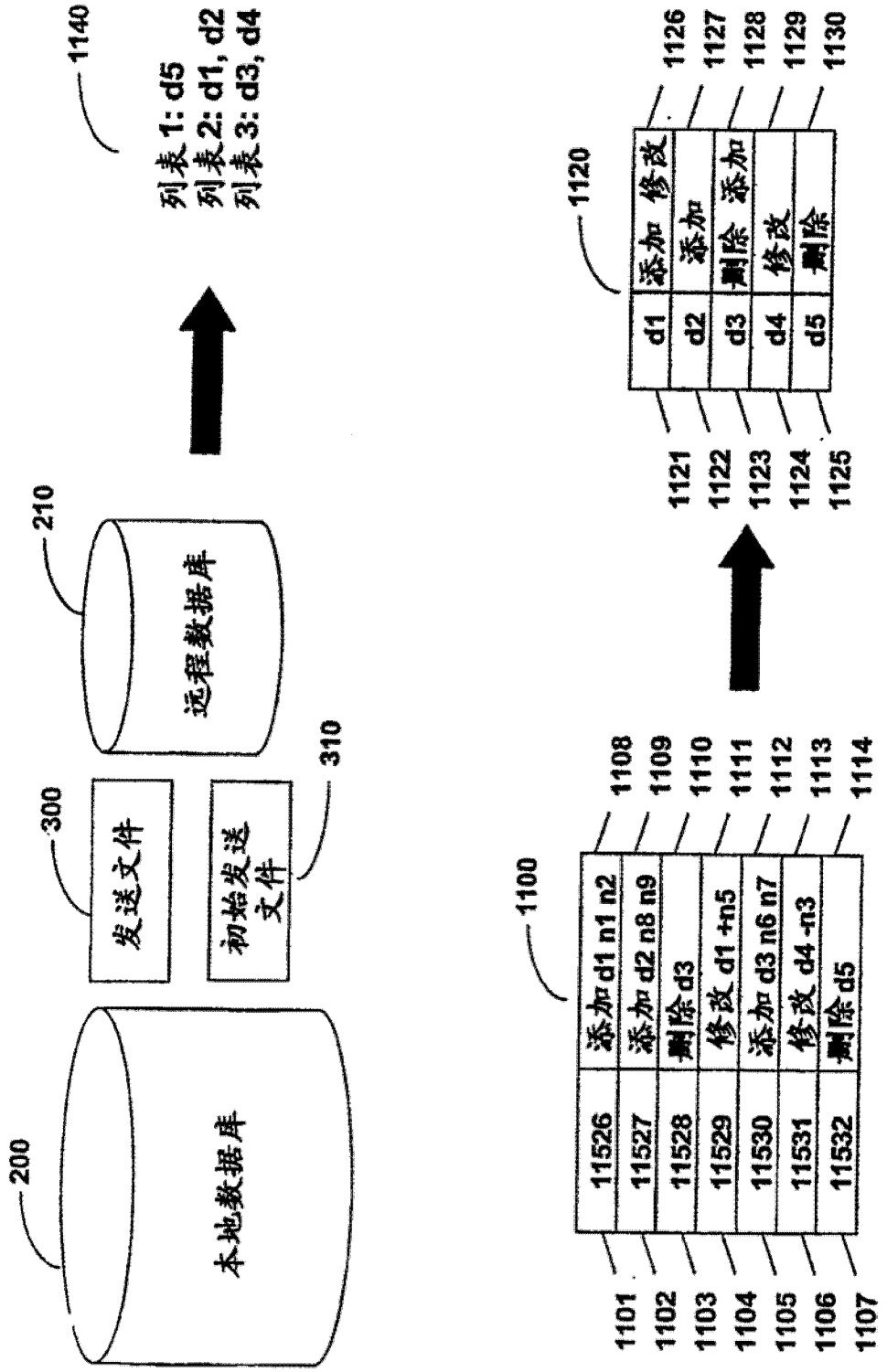

1140
列表1: d5
列表2: d1, d2
列表3: d3, d4
210
远程数据库
300
发送文件
310
初始发送文件
200
本地数据库
1120
d1 添加
d2 添加
d3 删除
d4 修改
d5 删除
修改 添加
1126 1127 1128 1129 1130
1121 1122 1123 1124 1125
1100
添加 d1 n1 n2
添加 d2 n8 n9
删除 d3
修改 d1 +n5
添加 d3 n6 n7
修改 d4 -n3
删除 d5
11526 11527 11528 11529 11530 11531 11532
1108 1109 1110 1111 1112 1113 1114
1101 1102 1103 1104 1105 1106 1107

图 11