



(12)发明专利

(10)授权公告号 CN 104094228 B

(45)授权公告日 2017.12.29

(21)申请号 201380008052.7

(22)申请日 2013.03.15

(65)同一申请的已公布的文献号
申请公布号 CN 104094228 A

(43)申请公布日 2014.10.08

(30)优先权数据

61/612,144 2012.03.16 US

61/774,356 2013.03.07 US

13/828,490 2013.03.14 US

13/828,590 2013.03.14 US

(85)PCT国际申请进入国家阶段日
2014.08.05

(86)PCT国际申请的申请数据
PCT/US2013/032473 2013.03.15

(87)PCT国际申请的公布数据
W02013/138774 EN 2013.09.19

(73)专利权人 甲骨文国际公司
地址 美国加利福尼亚

(72)发明人 P·帕金森

(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 11038
代理人 边海梅

(51)Int.Cl.
G06F 9/46(2006.01)

(56)对比文件
US 7284018 B1,2007.10.16,
US 6247023 B1,2001.06.12,
US 5680610 A,1997.10.21,
CN 102203779 A,2011.09.28,
CN 101341466 A,2009.01.07,
IBM Corp.IBM DB2 Universal Database -
Administration Guide: Planning -
Distributed Databases.《ftp://
public.dhe.ibm.com/ps/products/db2/info/
vr8/pdf/letter/db2dle80.pdf》.2002,171-
188.

审查员 郭妍捷

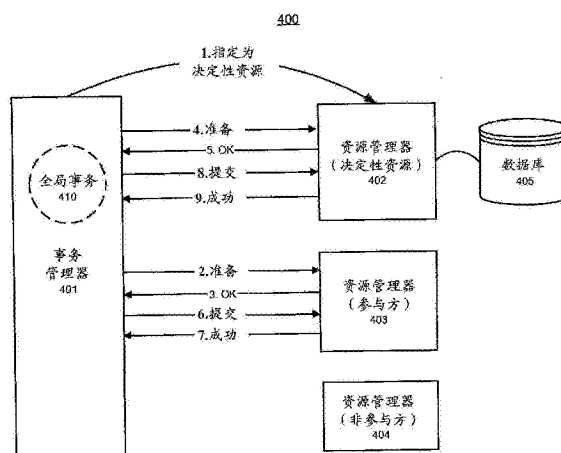
权利要求书3页 说明书7页 附图6页

(54)发明名称

用于支持基于二阶段提交调用的严格定序
的事务恢复的系统和方法

(57)摘要

提供用于支持基于二阶段提交调用的严格定序的事务恢复的系统和方法。为了在处理二阶段事务时支持消除中间层事务日志(TLOG),中间层事务型环境中的至少一个资源管理器可被指定为“决定性资源”。事务管理器可在决定性资源之前准备在中间层事务型系统中的所有其它的资源管理器。此外,事务管理器可依赖于由决定性资源为了恢复事务提供的要提交的未完成事务的列表。事务管理器可提交匹配从决定性资源返回的存疑事务的列表的、从资源管理器返回的存疑事务。否则,事务管理器可回滚存疑事务。



1. 一种用于恢复多个资源管理器上的事务的方法,该方法包括:

将所述多个资源管理器中的一资源管理器指定为决定性资源,其中所述决定性资源是在准备所述多个资源管理器中的所有其他资源管理器之后准备的并且是在提交所述多个资源管理器中的所有其他资源管理器之后提交的;

从所述决定性资源接收要提交的未完成事务的列表;

根据从所述决定性资源返回的存疑事务的列表构建/重新构建全局事务表GTT;

从所述多个资源管理器中的每个资源管理器检索存疑事务的列表,其中所述存疑事务是已经被准备但是还未提交的事务;

比较每个存疑事务与所述GTT;

在所述多个资源管理器中的至少一个资源管理器上提交存疑事务的列表中的与所述GTT匹配的一个或多个存疑事务,并且在所述多个资源管理器中的至少一个资源管理器上回滚不与所述GTT匹配的一个或多个存疑事务;以及

在对于所述多个资源管理器中的所有其他资源管理器成功提交或回滚了所述一个或多个存疑事务之后,提交从所述决定性资源检索的存疑事务的列表。

2. 根据权利要求1所述的方法,还包括:

允许所述事务是二阶段提交事务;以及

在执行所述二阶段提交事务中的准备阶段之前,将所述决定性资源持久保存在用于事务型中间层系统的配置中。

3. 根据权利要求1或2所述的方法,还包括:

允许所述多个资源管理器中的至少一个资源管理器是事务的非参与方。

4. 根据权利要求1或2所述的方法,还包括:

向所述决定性资源传递在所述事务中列出的所述多个资源管理器中的参与资源管理器的列表。

5. 根据权利要求4所述的方法,还包括:

配置所述决定性资源,以在恢复之前等待,直到所述参与资源管理器的列表被恢复。

6. 一种用于恢复事务的系统,该系统包括:

事务管理器;和

在中间层事务型系统中与所述事务管理器通信的多个资源管理器,并且

其中所述事务管理器进行操作以执行包括以下步骤的步骤:

将所述多个资源管理器中的一资源管理器指定为决定性资源,其中所述决定性资源是在准备所述多个资源管理器中的所有其他资源管理器之后准备的并且是在提交所述多个资源管理器中的所有其他资源管理器之后提交的;

从所述决定性资源接收要提交的未完成事务的列表;

根据从所述决定性资源返回的存疑事务的列表构建/重新构建全局事务表GTT;

从所述多个资源管理器中的每个资源管理器检索存疑事务的列表,其中所述存疑事务是已经被准备但是还未提交的事务;

比较每个存疑事务与所述GTT;

在所述多个资源管理器中的至少一个资源管理器上提交所述存疑事务的列表中的与所述GTT匹配的一个或多个存疑事务,并且在所述多个资源管理器中的至少一个资源管

理器上回滚不与所述GTT匹配的一个或多个存疑事务;以及

在对于所述多个资源管理器中的所有其他资源管理器成功提交或回滚了所述一个或多个存疑事务之后,提交从所述决定性资源检索的存疑事务的列表。

7. 根据权利要求6所述的系统,其中:

所述事务是二阶段提交事务;以及

所述事务管理器进行操作以在执行所述二阶段提交事务中的准备阶段之前将所述决定性资源持久保存在用于事务型中间层系统的配置中。

8. 根据权利要求6或7所述的系统,其中:

所述多个资源管理器中的至少一个资源管理器是所述事务的非参与方。

9. 根据权利要求6或7所述的系统,其中:

所述事务管理器进行操作以执行还包括以下步骤的步骤:

向所述决定性资源传递在所述事务中列出的所述多个资源管理器中的参与资源管理器的列表;和

配置所述决定性资源,以在恢复之前等待,直到所述参与资源管理器的列表被恢复。

10. 一种上面存储有指令的非暂时性机器可读存储介质,所述指令当被执行时使得系统执行包括以下步骤的步骤:

将多个资源管理器中的一资源管理器指定为决定性资源,其中所述决定性资源是在准备所述多个资源管理器中的所有其他资源管理器之后准备的并且是在提交所述多个资源管理器中的所有其他资源管理器之后提交的;

从所述决定性资源接收要提交的未完成事务的列表;

根据从所述决定性资源返回的存疑事务的列表构建/重新构建全局事务表GTT;

从所述多个资源管理器中的每个资源管理器检索存疑事务的列表,其中所述存疑事务是已经被准备但是还未提交的事务;

比较每个存疑事务与所述GTT;

在所述多个资源管理器中的至少一个资源管理器上提交存疑事务的列表中的与所述GTT匹配的一个或多个存疑事务,并且在所述多个资源管理器中的至少一个资源管理器上回滚不与所述GTT匹配的一个或多个存疑事务;以及

在对于所述多个资源管理器中的所有其他资源管理器成功提交或回滚了所述一个或多个存疑事务之后,提交从所述决定性资源检索的存疑事务的列表。

11. 一种用于恢复多个资源管理器上的事务的设备,该设备包括:

用于将所述多个资源管理器中的一资源管理器指定为决定性资源的装置,其中所述决定性资源是在准备所述多个资源管理器中的所有其他资源管理器之后准备的并且是在提交所述多个资源管理器中的所有其他资源管理器之后提交的;

用于从所述决定性资源接收要提交的未完成事务的列表的装置;

用于根据从所述决定性资源返回的存疑事务的列表构建/重新构建全局事务表GTT的装置;

用于从所述多个资源管理器中的每个资源管理器检索存疑事务的列表的装置,其中所述存疑事务是已经被准备但是还未提交的事务;

用于比较每个存疑事务与所述GTT的装置;

用于在所述多个资源管理器中的至少一个资源管理器上提交存疑事务的列表中的与所述GTT匹配的一个或多个存疑事务,并且在所述多个资源管理器中的至少一个资源管理器上回滚不与所述GTT匹配的一个或多个存疑事务的装置;以及

用于在对于所述多个资源管理器中的所有其他资源管理器成功提交或回滚了所述一个或多个存疑事务之后,提交从所述决定性资源检索的存疑事务的列表的装置。

12. 根据权利要求11所述的设备,还包括:

用于允许所述事务是二阶段提交事务的装置;以及

用于在执行所述二阶段提交事务中的准备阶段之前,将所述决定性资源持久保存在用于事务型中间层系统的配置中的装置。

13. 根据权利要求11或12所述的设备,还包括:

用于允许所述多个资源管理器中的至少一个资源管理器是事务的非参与方的装置。

14. 根据权利要求11或12所述的设备,还包括:

用于向所述决定性资源传递在所述事务中列出的所述多个资源管理器中的参与资源管理器的列表的装置。

15. 根据权利要求14所述的设备,还包括:

用于配置所述决定性资源,以在恢复之前等待,直到所述参与资源管理器的列表被恢复的装置。

用于支持基于二阶段提交调用的严格定序的事务恢复的系统和方法

[0001] 版权声明：

[0002] 本专利文件公开的一部分包含受到版权保护的材料。版权所有人不反对任何人传真复制专利文件或专利公开，只要它出现在专利和商标局的专利文件或记录中，但在其他方面无论如何保留所有版权权利。

技术领域

[0003] 本发明一般涉及诸如中间件的计算机系统和软件，特别是涉及事务型中间件。

背景技术

[0004] 事务型中间件系统或面向事务的中间件包括可处理组织内的各种事务的企业应用服务器。伴随诸如高性能网络和多处理器计算机的新技术的开发，需要进一步提高事务型中间件的性能。一般存在本发明的实施例意在针对的领域。

发明内容

[0005] 提供用于支持基于二阶段提交调用的严格定序的事务恢复的系统和方法。为了在处理二阶段事务时支持消除中间层事务日志 (TLOG)，中间层事务型环境中的至少一个资源管理器可被指定为“决定性 (determiner) 资源”。事务管理器可在决定性资源之前准备在中间层事务型系统中的所有其它的资源管理器。此外，事务管理器可依赖于由决定性资源为了恢复事务而提供的要被提交的未完成事务的列表。事务管理器可提交匹配从决定性资源返回的存疑事务的列表的、从资源管理器返回的存疑事务。否则，事务管理器可回滚存疑事务。

[0006] 对于本领域技术人员来说，当参照附图阅读各种实施例的以下详细描述时，本发明的其它目的和优点将变得十分明显。

附图说明

[0007] 图1表示根据本发明的实施例的事务型环境的示意图。

[0008] 图2表示在事务型中间件机器环境中支持二阶段提交的示意图。

[0009] 图3表示在事务型中间件机器环境中恢复存疑事务的示意图。

[0010] 图4表示根据本发明的实施例的用于在事务型中间件机器环境中支持用于处理事务的二阶段提交 (2PC) 调用的严格定序的示意图。

[0011] 图5表示根据本发明的实施例的用于在不使用事务日志 (TLOG) 的情况下恢复全局事务的示意图。

[0012] 图6是根据本发明的实施例的在不使用事务日志 (TLOG) 的情况下恢复二阶段提交事务的示例性的流程图。

具体实施方式

[0013] 在附图中作为例子而不是限制而图示本发明,在附图中,类似的附图标记表示类似的要素。应当注意,在本公开中提及的“一个”、“一种”或“一些”实施例未必指的是同一实施例,并且,这种提及意味着至少一个。

[0014] 以下的本发明的描述使用Tuxedo环境作为事务型中间件机器环境的例子。对于本领域技术人员来说,很显然,可没有限制地使用其它类型的事务型中间件机器环境。

[0015] 这里描述的是用于支持基于二阶段提交调用的严格定序的事务恢复的系统和方法。为了在处理二阶段事务时支持消除中间层事务日志(TLOG),中间层事务型环境中的至少一个资源管理器可被指定为“决定性资源”。事务管理器可在决定性资源之前准备在中间层事务型系统中的所有其它的资源管理器。此外,事务管理器可依赖于由决定性资源为了恢复事务提供的要被提交的未完成事务的列表。事务管理器可提交匹配从决定性资源返回的存疑事务的列表的、从资源管理器返回的存疑事务。否则,事务管理器可回滚(rollback)存疑事务。

[0016] 事务型环境和全局事务

[0017] 图1表示根据本发明的实施例的事务型环境的示图。如图1所示,事务型环境100可包含最终用户101、应用程序102、一个或更多个事务管理器(TM) 103、多个资源管理器(RM) 104~106以及一个或更多个持久性存储区,例如,数据库110。

[0018] 根据本发明的一个实施例,应用程序102可规定构成事务的动作。如图所示,应用程序102与事务管理器103通信以开始、提交或中止事务,并且,事务管理器103可向应用程序102送回事务的开始、结束和部署。此外,应用程序102可为可与多个资源管理器(RM) 104~106交换事务信息的事务管理器103限定事务边界。另外,应用程序102可经由嵌入结构化查询(SQL)与多个资源管理器104~106通信以完成有用的工作。

[0019] 多个资源管理器104~106可提供对例如为数据库110的持久性存储区的访问。根据本发明的实施例,多个资源管理器104~106可实现XA接口,以对持久性存储区透明地处理数据库连接和断开。XA接口可基于描述事务协调、提交和恢复的协议的规范。参与事务的XA资源可包含XA资源管理器和后端持久性存储区。

[0020] 根据本发明的各种实施例,事务型系统可支持可在多于一个的服务器上执行的全局事务,并且能够访问来自多于一个的资源管理器的数据。

[0021] 全局事务可被视为由原子性、一致性、隔离性和持续性(ACID)这四种性能表征的特定的一系列的操作。全局事务可以是具有以下特征的工作的逻辑单位:

- [0022] • 所有部分要么成功要么没有影响。
- [0023] • 执行正确地将资源从一个一致状态转换成另一个一致状态的操作。
- [0024] • 虽然同一事务中的其他处理可访问数据,但中间结果可能对于其他交易是不能访问的。
- [0025] • 任何类型的失败不能改变完成的序列的所有效果。

[0026] 并且,全局事务可包含几个局部事务,每个局部事务访问单个资源管理器。局部事务可访问单个数据库或文件,并且可被负责执行该相异的数据库处的更新的同时控制和原子性的资源管理器控制。给定的局部事务可能在完成其访问上是成功的或者不成功的。

[0027] 另外,事务管理器103可向事务型环境100中的不同的事务分配全局事务标识符(GTRID)。事务管理器103可监视它们的进程,并且负责确保事务完成并提供失败恢复。另外,事务管理器103可经由XA接口与多个资源管理器104~105通信,以交换事务信息,诸如向资源管理器104~105发送二阶段提交调用。

[0028] 二阶段提交(2PC)

[0029] 二阶段提交(2PC)协议可被用于执行诸如松散耦合的全局事务的事务。二阶段提交协议(2PC)可包含准备阶段和提交阶段。在准备阶段中,协调事务管理器(TM)指示参与资源管理器(RM)对提交或中止事务采取必要的步骤。在提交阶段中,事务管理器(TM)基于准备阶段的结果决定是提交还是中止事务。

[0030] 图2表示在事务型中间件机器环境中支持二阶段提交的示图。如图2所示,事务型环境200可包括支持各种事务的执行的的事务管理器201和管理一个或多个例如为数据库205的数据源的一个或多个资源管理器202~204。

[0031] 例如,事务管理器201可执行包含事务分支A211、事务分支B212和事务分支C213的事务,每个事务分支可分别针对资源管理器202~204被执行。如果任何分支在事务中失败,那么事务管理器201可帮助资源管理器202~204决定是提交还是回滚事务。

[0032] 如图2所示,事务管理器201可在所有三个分支上向资源管理器202~204发送准备指令(步骤1、2和3)。在资源管理器202~204返回“OK”表决(步骤4、5和6)之后,事务管理器201可将事务日志写入到数据库205(步骤7)。

[0033] 事务日志(TLOG)可被写入到文件或数据库,使得事务管理器201可在任何分支在提交阶段期间失败的情况下具有足够的信息以恢复事务。

[0034] 然后,事务管理器201可指示资源管理器202~204提交所有的三个分支(步骤8、9和10)。资源管理器202~204可在成功完成提交阶段之后通知事务管理器201(步骤11、12和13)。

[0035] 基于事务日志(TLOG)的事务恢复

[0036] 根据本发明的一个实施例,事务日志(TLOG)可保持用于通过事务管理器提交事务的决定。例如,TLOG可存储资源检查点记录,该资源检查点记录可被事务管理器持久保存,以使其能够跟踪不同的参与XA资源。

[0037] 事务可在事务管理器在准备阶段之后从所有事务分支接收成功表决之后被写入TLOG。TLOG中的事务记录可至少包含事务标识符(XID),该事务标识符又包含由事务管理器分配的GTRID和执行事务的资源管理器分配的本地XID。

[0038] 此外,TLOG可保持可被标记为要被提交的进行中事务的状态的记录。TLOG对于在系统崩溃之后恢复存疑事务是有益的,这些存疑事务是已被准备但还没有在中间层事务型系统中提交的事务。在不恢复存疑事务的情况下,系统可能在崩溃之后处于不正确和不一致的状态中。

[0039] 例如,如果系统在二阶段提交事务期间失败,那么对一个后端数据存储区的更新可能已被提交,但是在同一事务中对另一数据存储区的更新还没有被指令提交,即,数据存储区的更新仍是未决的。一旦系统的失败部分被重新启动,保持未决的更新的数据存储区就可能不能获知更新是否应被提交还是回滚。

[0040] 图3表示在事务型中间件机器环境中恢复存疑事务的示图。如图3所示,事务型中

间件机器环境300包含事务管理器301和多个资源管理器302~304以及持久性存储区即数据库305。

[0041] 事务管理器301可通过读取TLOG自动确定全局事务是否存疑(步骤1)。然后,事务管理器301可轮询参与资源管理器302~304的有关后端数据存储区。参与资源管理器302~304中的每一个可返回存疑事务的列表,这些存疑事务是事务管理器301不知道是要回滚它还是要提交的事务(步骤2、4和6)。

[0042] 此外,事务管理器301可匹配每个存疑事务与TLOG,并且前进以在不同的资源管理器302~304上提交或回滚存疑事务(步骤3、5和7)。例如,当在TLOG中出现存疑事务时,XAResource.commit()可被调用,以向在TLOG中指定的资源管理器提交存疑事务。另一方面,当事务不处于TLOG中,即在崩溃之前还没有在事务上做出提交决定时,XAResource.rollback()可被调用,以基于事务标识符(XID)在资源管理器上回滚它。

[0043] 如图3所示,事务可在资源管理器302被准备,并且系统在资源管理器302可向事务管理器301发送成功表决之前崩溃。然后,事务管理器301可能不能从所有的事务分支接收成功表决,并因此不在TLOG中记录事务。因此,事务管理器301可在它们的相应的资源管理器302~304上回滚所有的分支事务。因此,通过使用这种一致且可预测的事务恢复方法,事务管理器可避免一些分支被提交而一些分支被回滚的混合的启发式完成。

[0044] 避免需要事务管理器恢复日志

[0045] 在事务型环境中,为了进行恢复并确保ACID性能是二阶段提交事务管理器的设计要求,恢复信息可持久保存在稳定存储设备中。

[0046] 例如,(XA)二阶段记录算法可包括以下步骤:

[0047] 1.应用在事务中使用多个XA资源并且对于事务发出提交。

[0048] 2.事务管理器在所有XA资源参与方上发出XAResource.prepare。

[0049] 3.事务管理器持久保存包含事务的格式id和全局事务id的事务日志。

[0050] 4.事务管理器在所有XA资源参与方上发出XAResource.commit。

[0051] 5.事务管理器(一般懒惰地、成批地,等等)清除/删除事务记录。

[0052] 6.在失败的情况下,持久保存的事务记录被与从在所有涉及的资源管理器上做出的XAResource.recover调用返回的存疑Xid进行匹配,并因此进行通过提交或回滚的恢复。

[0053] 需要将恢复信息连同对于资源管理器的实际协议网络调用持久保存在稳定存储设备中可能增加二阶段提交事务的执行成本。它还可承担由于高度可用稳定存储设备所需要的共享文件系统或数据库存储设备导致的资产成本以及由于该存储设备的管理导致的管理成本。这些成本在需要(同步)复制及其重新管理的站点大范围灾难恢复的情况下会升级。因此,避免需要事务管理器恢复日历与现有的系统相比可提供明显的改善。

[0054] 根据本发明的一个实施例,通过使用以下的技术,通过基于二阶段提交调用的严格定序和一个资源参与方被分类作为恢复的决定性来提供事务恢复和ACID性能,能够避免对事务日志的需要:

[0055] 1.应用在事务中使用多个XA资源,并且对事务发出提交。

[0056] 2.某些资源可被配置为被指定为事务结果的“决定性”的。如果没有这种资源被配置,那么事务系统将提名一个并在配置中持久保存它。

[0057] 3.无论什么时候在二阶段事务中首次使用资源或者首次使用资源的新组合,都通

过事务处理子系统自动地捕获该组合并且将其持久保存在配置中。该配置持久保存紧挨着对这种事务发出的首次准备之前发生。如果这种配置改变失败,那么事务可被回滚。虽然不是必须的,但该步骤提供进一步的优化,原因是它减少在恢复期间需要XAResource.recover调用的资源组。

[0058] 4.事务管理器放弃(forgo)任何事务记录。

[0059] 5.事务管理器在所有XA资源参与方上发出XAResource.prepare,最后准备“决定性”资源。

[0060] 6.事务管理器在所有XA资源参与方上发出XAResource.commit,最后提交“决定性”资源。

[0061] 7.在失败的情况下,在配置中的所有非“决定性”资源上进行XAResource.recover扫描,并且,存疑Xid被与从决定性资源的XAResource.recover返回的存疑Xid相匹配。因此,如果在决定性列表中存在匹配/现有Xid,则通过提交恢复,或者,如果不存在,则通过回滚恢复。

[0062] 8.在事务系统的干净关闭或重新开始期间,配置以不在启动期间进行不必要的恢复调用和处理的方式被清除/标记。

[0063] 在以下的部分中更详细地描述上述的技术。

[0064] 二阶段提交(2PC)调用的严格定序

[0065] 根据本发明的实施例,系统可基于二阶段提交调用的严格定序消除处理事务中的中间层事务日志(TLOG)。

[0066] 图4表示根据本发明的实施例的用于在事务型中间件机器环境中支持用于处理事务的二阶段提交(2PC)调用的严格定序的示图。如图4所示,可在包含事务管理器(TM)401、多个资源管理器(RM)402~404的中间层事务型系统400中支持全局事务410。资源管理器(RM)402~403参与全局事务410,而资源管理器404不参与全局事务410。根据本发明的实施例,事务410不跨过多个事务管理器。

[0067] 事务管理器401可将例如为与数据库405相关联的资源管理器(RM)402的资源指定为决定性资源(步骤1)。可在用于事务机器环境的配置中持久保存对资源管理器(RM)402为决定性资源的指定。当在配置中在事务中没有决定性资源被指定时,事务管理器401可提名数据源为决定性资源,并在配置中持久保存它。另外,决定性资源402可以是数据库以外的源,诸如消息队列。

[0068] 此外,无论什么时候在处理二阶段事务410时首次使用资源或者首次使用资源的新组合,都可自动地通过事务管理器401捕获新资源或新资源组合并且将其持久保存在诸如配置文件的配置中。该配置持久保存可在紧挨着对二阶段事务发出首次准备请求之前发生。如果该配置改变失败,那么事务410可被回滚。

[0069] 根据本发明的实施例,事务管理器401可确保对于二阶段事务410,在所有参与资源管理器中最后准备和提交例如为资源管理器402的决定性资源。

[0070] 如图4所示,事务管理器401可在准备决定性资源(步骤4)之前首先准备资源管理器(RM)403(步骤2)。在从参与资源管理器(RM)402~403接收到“OK”表决(步骤3和5)之后,事务管理器401可提交资源管理器(RM)403(步骤6),然后是决定性资源(步骤8)。事务可在事务管理器401从资源管理器402~403中的每一个接收到成功指示(步骤7和9)之后完成。

[0071] 因此,事务管理器401可放弃资源/检查点的任何记录,包含在成功处理准备阶段之后写入TLOG。系统可在允许中间层事务型系统400恢复事务410的同时改善事务的性能。

[0072] 在不使用TLOG的情况下恢复全局事务

[0073] 图5表示根据本发明的实施例的用于在不使用事务日志(TLOG)的情况下恢复全局事务的示意图。如图5所示,可在包含事务管理器(RM) 501、多个资源管理器(RM) 502~504的中间层事务型系统500中支持全局事务510。

[0074] 当事务510需要被恢复时,事务管理器501可尝试例如通过在决定性资源502上放置XA_recover()调用恢复(步骤1)。事务管理器501可从决定性资源502接收存疑事务(准备但未提交的事务)的列表(步骤2)。此外,事务管理器501可使用存疑事务的列表以构建/重新构建全局事务表(GTT506)(步骤3)。

[0075] 根据本发明的一个实施例,当在二阶段提交事务中成功准备决定性资源502时,存疑事务的列表可与事务管理器501已经指示参与资源管理器502~503提交的中间层未完成事务相同。

[0076] 然后,事务管理器501可尝试例如通过在包含参与资源管理器503(步骤9)和非参与资源管理器504(步骤10)在内的事务型中间层系统500中的所有其它资源管理器上放置XA_recover()调用恢复(步骤10)。

[0077] 例如,事务管理器501可从参与资源管理器503接收存疑事务的列表(步骤5)。事务管理器501可匹配列表中的存疑事务与GTT506表以产生恢复列表(步骤6和7)。然后,事务管理器501可基于恢复列表恢复参与资源管理器503(步骤8)。例如,事务管理器501可提交匹配GTT506表的存疑事务,并且可回滚不匹配GTT506表的存疑事务。

[0078] 以上过程可在作为非参与资源的资源管理器504上执行。如图5所示,在于资源管理器504上放置XA_recover()调用之后,事务管理器501可接收存疑事务的列表(步骤11),这些存疑事务可被与GTT506相匹配以产生恢复列表(步骤12)。然后,事务管理器501可在资源管理器504上提交/回滚存疑事务(步骤14和15)。

[0079] 一旦恢复所有其它资源管理器503~504(步骤9和15),事务管理器501就可向决定性资源管理器提交准备的事务的列表中的所有事务(步骤16和17)。最后,事务管理器501可去除GTT506中的条目(步骤18)。

[0080] 如这里描述的那样,即使一个或更多个资源管理器不是事务的一部分,在所有其它资源管理器503~504被恢复之前,来自存疑事务的列表的事务不可被提交到决定性资源管理器502。

[0081] 在本发明的一个实施例中,如果在准备阶段中可传递关于参与资源502、503的附加信息并将其持久保持在决定性资源502中,那么事务管理器501可知道事务的参与资源502、503。通过该附加信息,事务管理器501可不在等待所有已知资源管理器502~504恢复的情况下,提交从决定性资源管理器502返回的存疑事务的列表中的事务。事务管理器501可能只需要等待,直到事务所有参与方502~503被恢复。

[0082] 另外,在事务型中间层系统500的干净关闭或重新开始期间,配置可按可在启动期间避免不必要的恢复和处理的方式被清除或标记。

[0083] 图6是根据本发明的实施例的在不使用事务日志(TLOG)的情况下恢复二阶段提交事务的示例性的流程图。如图6所示,在步骤601中,事务管理器可在决定性资源上执行XA_

recover() 调用,并接收要在中间层事务型系统中提交的未完成事务的列表。然后,在步骤602处,事务管理器可使用准备的事务的列表以重新构建全局事务表(GTT)表。接下来,在步骤603处,事务管理器可在包含事务的参与方或非参与方在内的系统中的其它资源管理器中的每一个上执行XA_recover()调用,并且可从其它资源管理器中的每一个接收存疑事务的列表。在步骤604处,事务管理器可通过匹配事务与GTT表从存疑事务的列表恢复每个事务。如果在GTT表中找到匹配,则事务可被提交到可从中检索该事务的资源管理器;否则,它可被回滚。最后,在步骤605处,事务管理器可在其它资源管理器上的所有存疑事务恢复之后提交来自决定性资源的存疑事务中的事务。

[0084] 可通过使用一个或更多个常规的通用或专用数字计算机、计算装置、机器或微处理器,包括根据本公开的教导编程的一个或更多个处理器、存储器和/或计算机可读存储介质在内,来方便地实现本发明。熟练的编程人员可很容易地基于本公开的教导准备适当的软件编码,这对于软件领域技术人员来说是十分明显的。

[0085] 在一些实施例中,本发明包括计算机程序产品,该计算机程序产品是上面/其中存储有可被用于将计算机编程以执行本发明的处理中的任一个的指令的存储介质或计算机可读介质(媒体)。存储介质可包括但不限于:任何类型的盘,包含软盘、光盘、DVD、CD-ROM、微驱动和磁光盘、ROM、RAM、EPROM、EEPROM、DRAM、VRAM、闪存存储器装置、磁卡或光卡、纳米系统(包含分子存储器IC)或适于存储指令和/或数据的任何类型的介质或装置。

[0086] 已出于图示和描述的目的提供了本发明的以上描述。它意图不在于详尽或者将本发明限于公开的确切的形式。对于本领域技术人员来说,许多修改和变化是十分明显的。为了最佳地解释本发明的原理及其实际应用,选择和描述了各实施例,由此使得本领域技术人员能够对于各种实施例以及通过适于设想的特定用途的各种修改理解本发明。本发明的范围要由以下的权利要求以及它们的等同物限定。

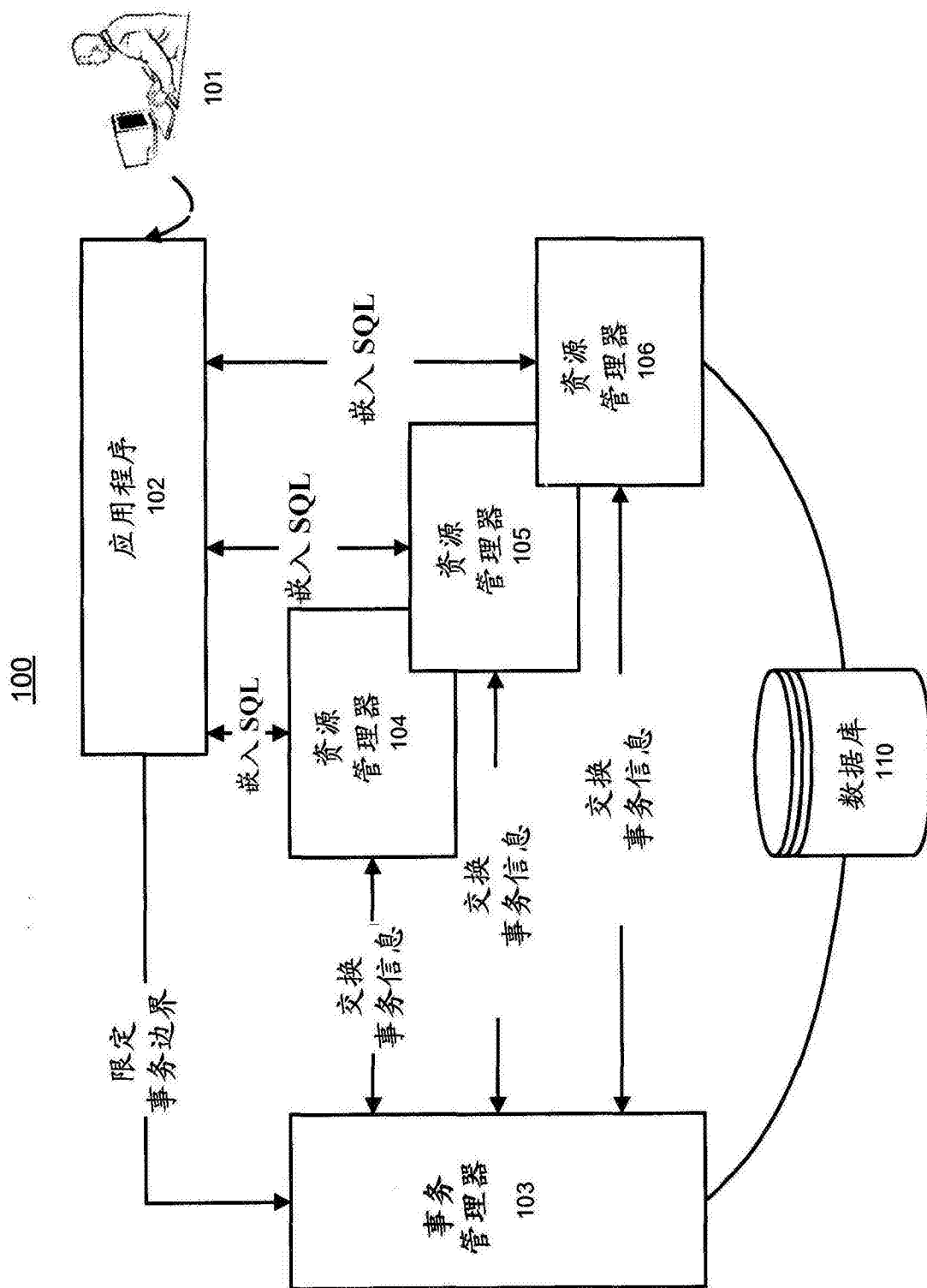


图1

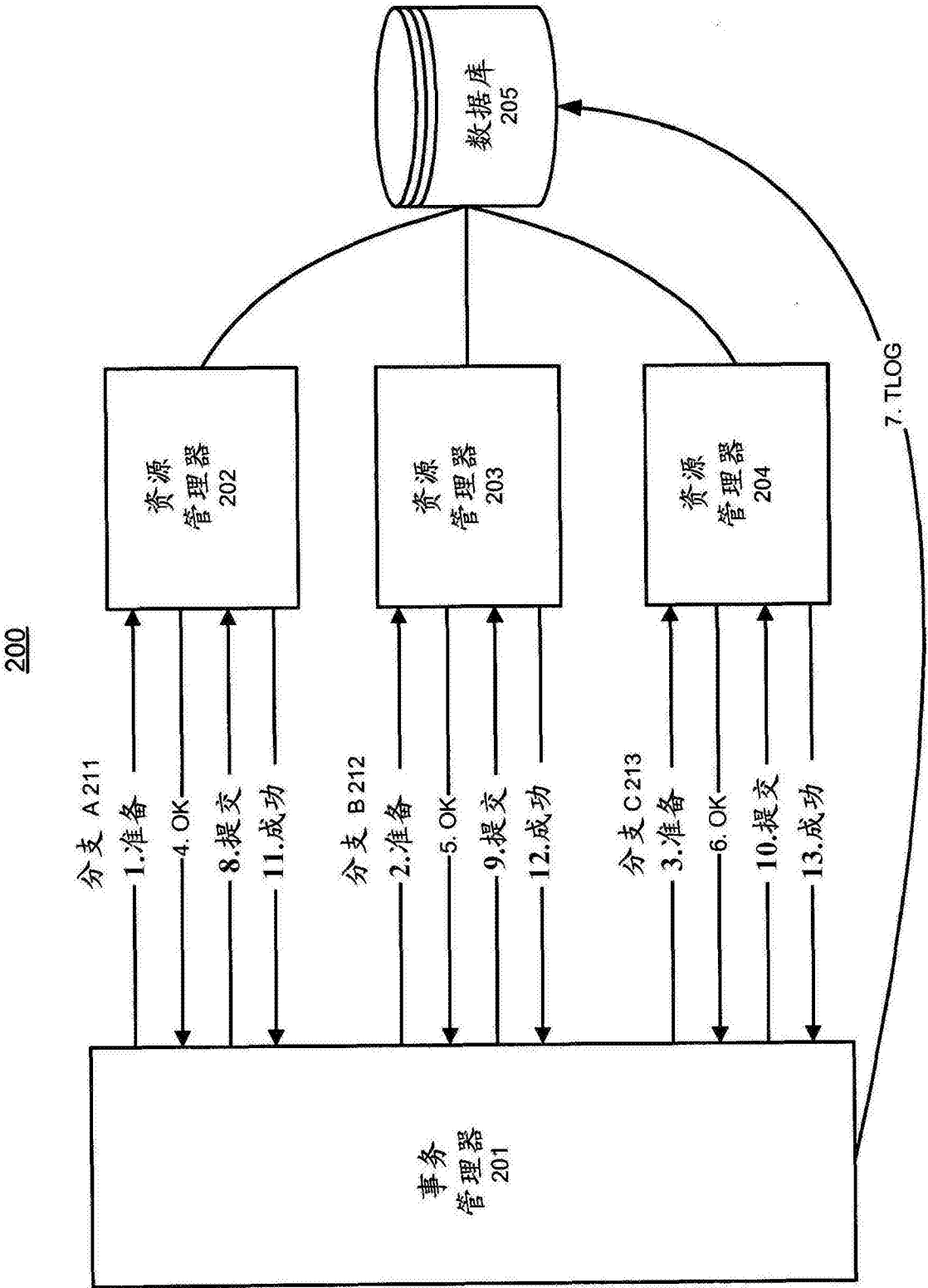


图2

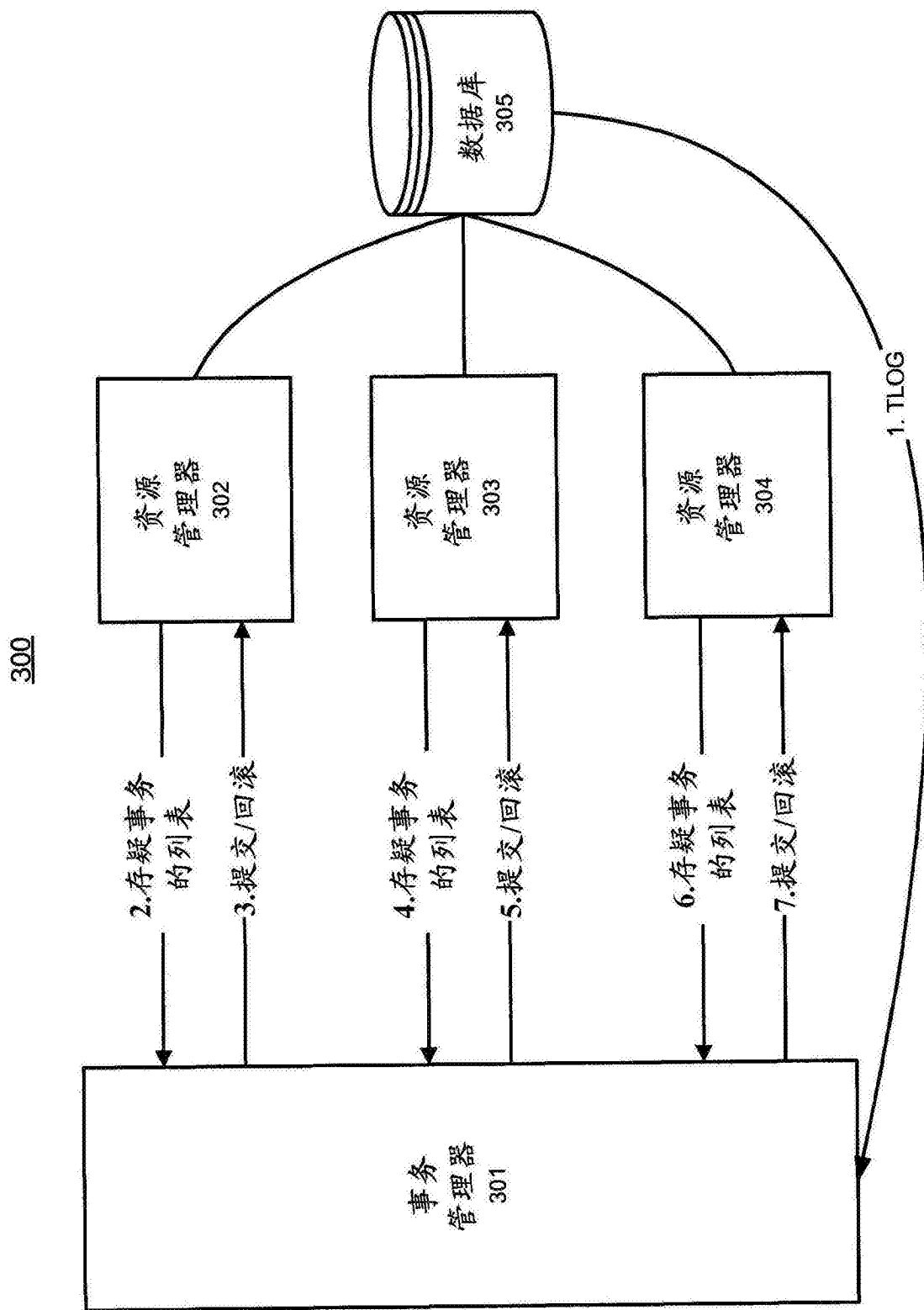


图3

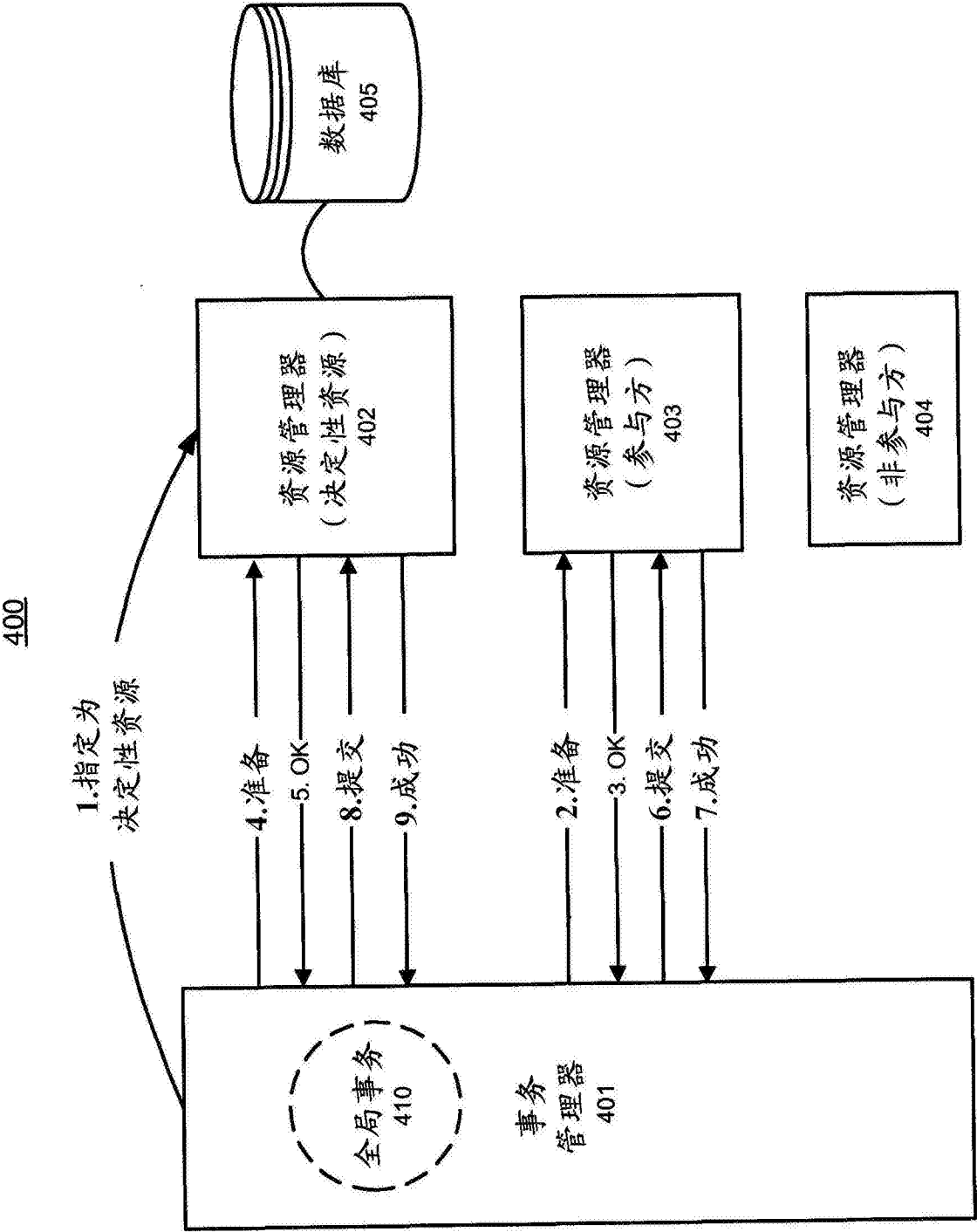


图4

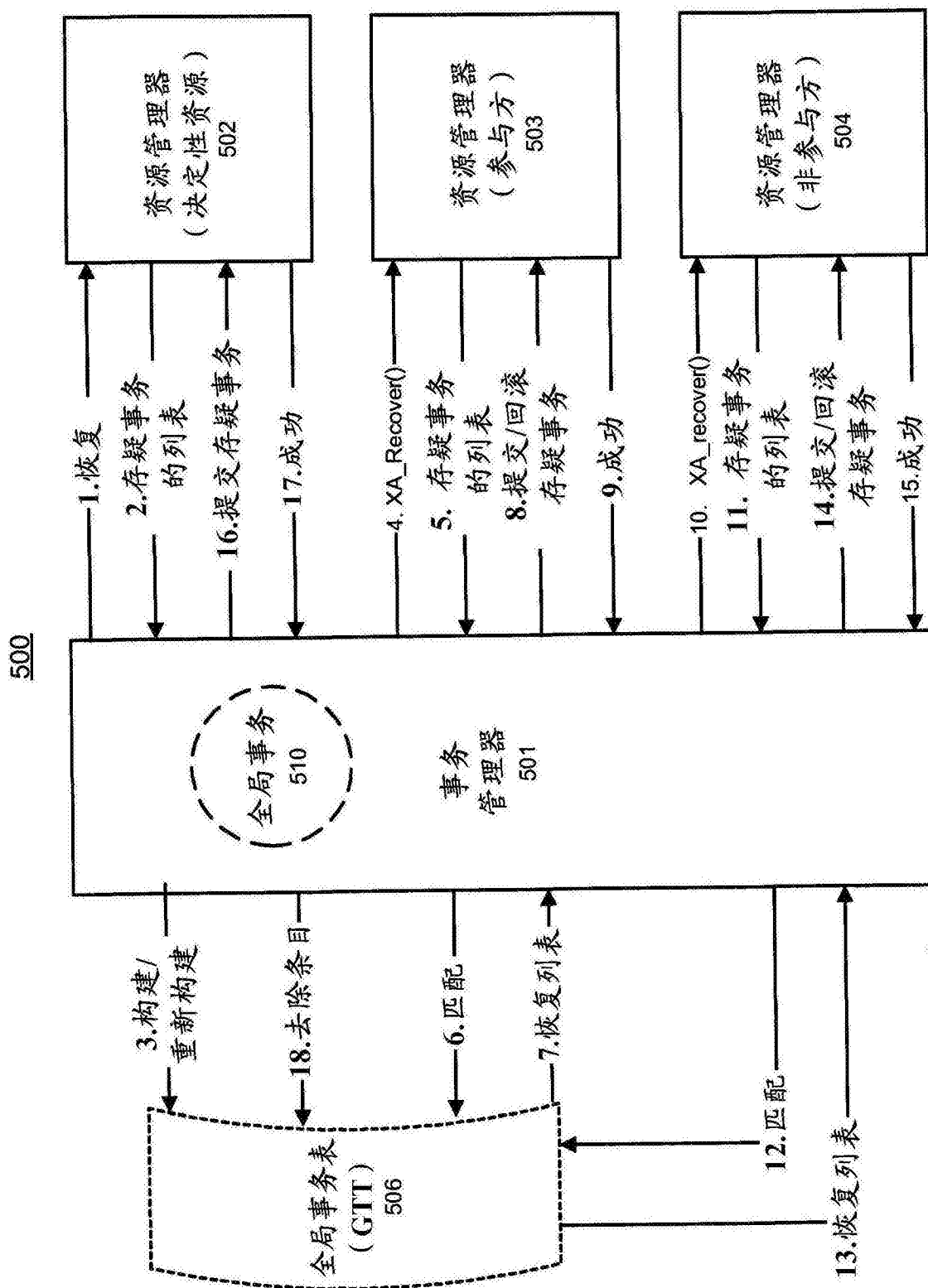


图5

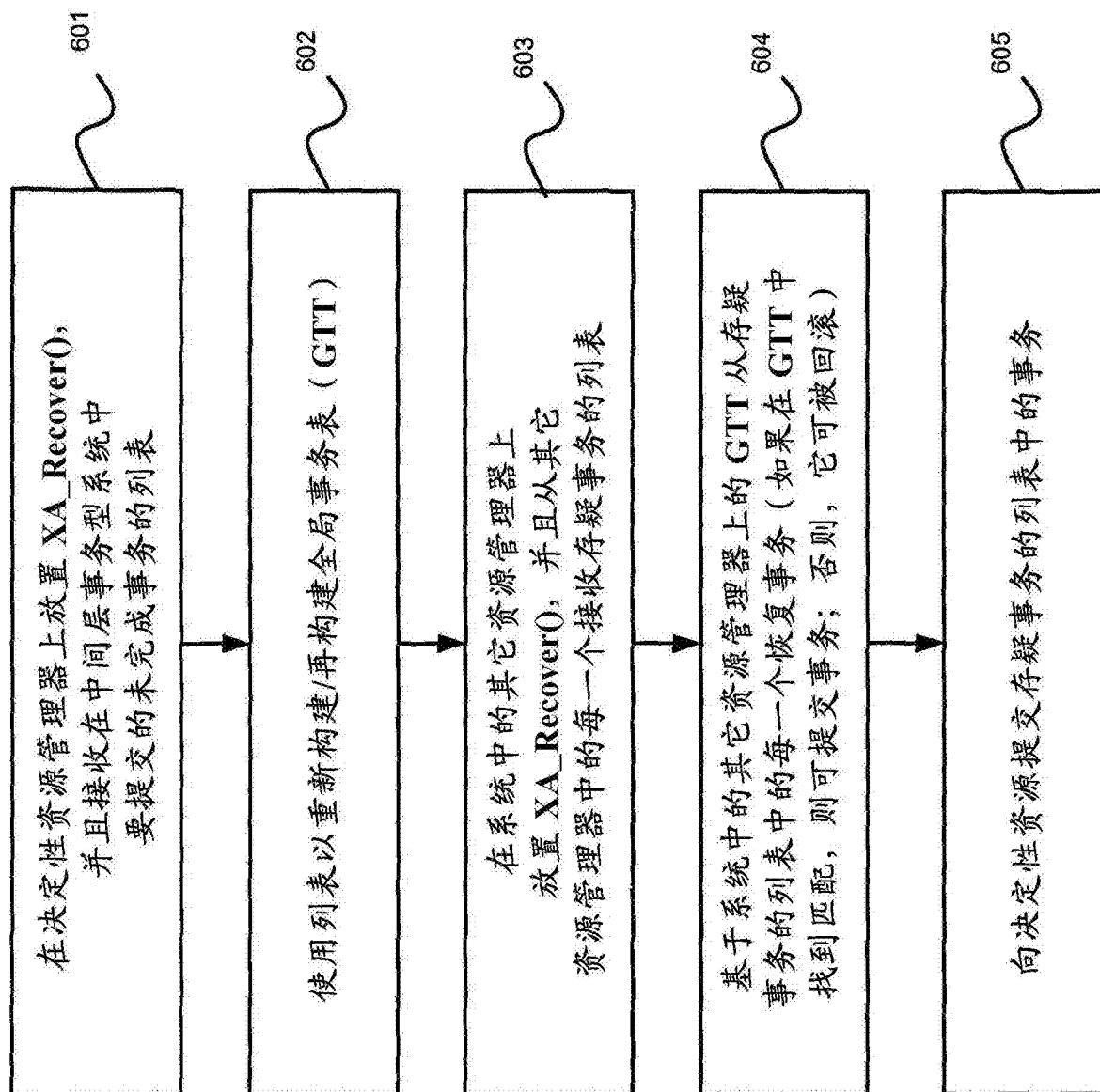


图6