

(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国际局

(43) 国际公布日
2018 年 1 月 4 日 (04.01.2018)



(10) 国际公布号
WO 2018/001135 A1

(51) 国际专利分类号:
G06F 17/30 (2006.01) **G06F 12/08** (2016.01)

(21) 国际申请号: PCT/CN2017/089035

(22) 国际申请日: 2017 年 6 月 19 日 (19.06.2017)

(25) 申请语言: 中文

(26) 公布语言: 中文

(30) 优先权:
201610496073.3 2016 年 6 月 30 日 (30.06.2016) CN

(71) 申请人: 华为技术有限公司 (HUAWEI TECHNOLOGIES CO., LTD.) [CN/CN]; 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。

(72) 发明人: 彭勇飞 (PENG, Yongfei); 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。

(74) 代理人: 北京龙双利达知识产权代理有限公司 (LONGSUN LEAD IP LTD.); 中国北京市海淀区北清路 68 号院 3 号楼 101, Beijing 100094 (CN)。

(81) 指定国 (除另有指明, 要求每一种可提供的国家保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL,

(54) Title: METHOD FOR PROCESSING DATABASE TRANSACTION, CLIENT AND SERVER

(54) 发明名称: 处理数据库事务的方法、客户端和服务端

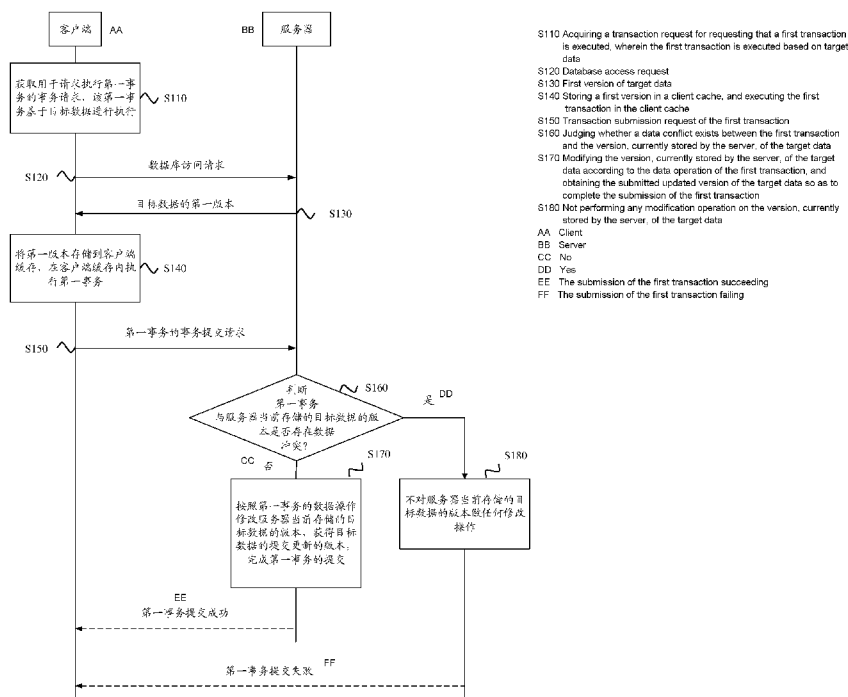


图 2

(57) Abstract: A method for processing a database transaction, a client and a server. The method comprises: a server receiving a database access request, sent by a client, for requesting for target data (S120); sending a first version of the target data to the client (S130), wherein the first version is the version, currently stored by the server, of the target data, and the server only saves the most recently submitted updated version of the target data; receiving a transaction submission request sent by the client after executing a first transaction based on the first version (S150), wherein the transaction submission request comprises a log for recording a data operation

SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG,
US, UZ, VC, VN, ZA, ZM, ZW。

- (84) 指定国 (除另有指明, 要求每一种可提供的地区
保护): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ,
NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚 (AM,
AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG,
CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU,
IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT,
RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI,
CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG)。

本国际公布:

- 包括国际检索报告 (条约第21条(3))。

of the first transaction; and according to the log, where it is determined that a data conflict does not exist between the data operation of the first transaction and the version, currently stored by the server, of the target data, modifying the version, currently stored by the server, of the target data according to the data operation of the first transaction, and obtaining the submitted updated version of the target data (S170).

(57) 摘要: 一种处理数据库事务的方法、客户端与服务器, 该方法包括: 服务器接收客户端发送用于请求目标数据的数据库访问请求 (S120); 向客户端发送目标数据的第一版本 (S130), 第一版本为服务器当前存储的目标数据的版本, 服务器中只保存目标数据的最近一次提交更新的版本; 接收客户端基于第一版本执行完第一事务后发送的事务提交请求 (S150), 事务提交请求中包括用于记录第一事务的数据操作的日志; 根据日志, 在确定第一事务的数据操作与服务器当前存储的目标数据的版本不存在数据冲突的情况下, 按照第一事务的数据操作修改服务器当前存储的目标数据的版本, 获得目标数据的提交更新的版本 (S170)。

处理数据库事务的方法、客户端和服务端

本申请要求于 2016 年 6 月 30 日提交中国专利局、申请号为 201610496073.3、发明名称为“处理数据库事务的方法、客户端和服务端”的中国专利申请的优先权，其全部内容通过引用结合在本申请中。

技术领域

本申请涉及数据库技术领域，并且更具体地，涉及一种处理数据库事务的方法、客户端和服务端。

背景技术

当多个用户同时访问数据库时，会产生多个事务并发操作同一数据的情况，若对这种并发操作不加控制就可能会产生在数据库中存取不正确数据的问题，破坏事务与数据库的一致性。

封锁是实现并发控制的一个非常重要的技术。所谓封锁就是事务 T 对某个数据操作之前，先向系统发出对其加锁的请求，加锁后事务 T 就对该数据有了一定的控制，在事务 T 释放它的锁之前，其他事务均不能对该数据进行操作。封锁对象的大小称为封锁粒度。封锁对象可以是逻辑单元，也可以是物理单元。上述封锁的技术也称为锁机制 (Lock)。

在锁机制中，读会阻塞写，写也会阻塞读，当封锁粒度较大时，会降低数据库的访问实时性，导致修改同一数据时阻塞较大。而且同一数据的读写冲突会产生功能长时间阻塞，容易产生死锁问题，需要定期做死锁检测。

针对锁机制存在的问题，当前技术提出了多版本并发控制 (Multiversion Concurrency Control, MVCC) 的事务机制。一般把基于锁的并发控制机制称为悲观机制，而把 MVCC 机制称为乐观机制。

在 MVCC 中，为同一数据维护多个版本 (每个版本对应一个时间戳)，即这个数据被修改提交一次就生成一个新的版本，同一数据的多个版本之间的串行化使用时间戳 (或版本号) 或事务标识 ID 的方式来支持。对于要对该数据进行读操作的事务 T 来说，总是可以提供时间戳与事务 T 的开始时刻相一致的版本，不管这个事务 T 执行的时间有多长，该事务 T 始终看到的是该数据的同一个版本，即使在该事务 T 执行期间该数据被其他事务修改了，该事务 T 也不会出现“脏”读或幻读的现象。因此，MVCC 通过为同一数据维护多个版本，实现了读操作与写操作并发而互不影响，也可以支持多个读操作并发而互不影响。但是，为同一数据维护多个版本的方式并不能解决并发的写操作与写操作之间的冲突。在 MVCC 中，还是利用加锁的方式解决来写操作的冲突问题。如果一个事务对一个数据执行写操作，该数据就被锁定，在该事务释放它的锁之前，其他事务均不能对该数据进行操作，即多个写操作会串行执行，因此不会存在冲突。

上述可知，现有的 MVCC 机制相对于传统的锁机制，避免了读阻塞写，写阻塞读，但是还是无法避免写阻塞写。此外，在现有的 MVCC 技术中，由于需要存储同一数据的多个版本，导致数据库的存储管理较为复杂，而且还需要定期清理不再需要的版本以回收空间，也增加了额外的开销。

发明内容

本申请提供了一种处理数据库事务的方法、客户端和服务端，实现了一种能够支持多个读操作和/或写操作同时并发的无锁事务机制，并且能够降低数据库的存储管理复杂度。

5 第一方面提供了一种处理数据库事务的方法，该方法包括：服务器接收客户端发送的数据库访问请求，所述数据库访问请求用于请求目标数据；所述服务器基于所述数据库访问请求，向所述客户端发送所述目标数据的第一版本，所述第一版本为所述服务器当前存储的所述目标数据的版本，所述服务器只保存所述目标数据的最近一次提交更新的版本；所述服务器接收所述客户端基于所述第一版本执行完第一事务后发送的事务提交请求，所述事务提交请求用于请求提交所述第一事务，所述事务提交请求中包括用于记录所述第一事务的数据操作的日志；所述服务器根据所述日志，在确定所述第一事务的数据操作与所述服务器当前存储的所述目标数据的版本不存在数据冲突的情况下，按照所述第一事务的数据操作修改所述服务器当前存储的所述目标数据的版本，获得所述目标数据的提交更新的版本，以完成所述第一事务的提交。

15 在本申请技术方案中，所述服务器只保存所述目标数据的最近一次提交更新的版本，因此，所述服务器当前存储的所述目标数据的版本自然是所述目标数据的最近一次提交更新的版本。

本申请中描述的所述服务器当前存储的所述目标数据的版本是动态的概念，例如在时刻 t1，所述服务器当前存储的所述目标数据的版本指的是 t1 对应的当前时刻所述服务器存储的所述目标数据的版本；在时刻 t2，所述服务器当前存储的所述目标数据的版本指的是 t2 对应的当前时刻所述服务器存储的所述目标数据的版本。

所述目标数据的最近一次提交更新的版本指的是所述目标数据的最近一次经过要提交的事务的修改操作之后的版本，即所述目标数据的最近一次提交更新的版本满足事务与数据库的一致性。

25 所述第一事务的数据操作与所述服务器当前存储的所述目标数据的版本不存在数据冲突指的是，所述第一事务的数据操作与所述服务器当前存储的所述目标数据的版本的最近一次修改操作之间没有冲突，使得按照第一事务的数据操作修改所述服务器当前存储的目标数据的版本，不会破坏事务与数据库的一致性。

在本申请技术方案中，服务器接收到客户端发送的请求目标数据的数据库访问请求后，通过向客户端发送目标数据的第一版本，该第一版本为服务器当前存储的目标数据的版本，且该版本为目标数据的最近一次提交更新的版本，使得可以在客户端本地缓存中基于该第一版本完成事务的执行；在接收到客户端发送的事务提交请求时，在确定要提交的事务的数据操作与服务器中当前存储的该目标数据的版本不存在数据冲突的情况下，按照要提交的事务的数据操作修改服务器当前存储的该目标数据的版本，获得该目标数据的提交更新的版本，以完成事务的提交。应理解，通过向客户端发送目标数据的第一版本，使得在客户端执行事务，能够保证并发事务之间的隔离性，即实现了并发的读操作之间没有影响，并发的读操作与写操作之间也没有影响，并发的写操作与写操作之间也没有影响。在服务器提交事务，并在确定待提交的事务的数据操作与服务器当前存储的目标数据的版本不存在数据冲突的情况下，才执行事务的提交，能够保证事务与数据库的一致性。此外，通过在客户端基于目标数据的第一版本执行事务，使得服务器

上对该目标数据无需维护多个版本，只需维护该目标数据的最近一次提交更新的版本即可，相对于现有的 MVCC 技术，本申请能够有效降低数据库的存储管理复杂度，从而减小服务器的存储负担。因此，本申请提供了一种能够支持多个读操作和/或写操作同时并发的无锁事务机制，并且能够有效降低数据库的存储管理复杂度。

5 结合第一方面，在第一方面的第一种可能的实现方式中，所述第一事务的数据操作包括插入操作，所述日记中包括所述插入操作对应的主关键字（Primary Key），所述方法还包括：在所述服务器当前存储的所述目标数据的版本中不存在所述主关键字的情况下，所述服务器确定所述第一事务的数据操作与所述服务器当前存储的所述目标数据的版本不存在数据冲突。

10 具体地，所述服务器当前存储的所述目标数据的版本中不存在所述插入操作对应的主关键字指的是，所述服务器当前存储的所述目标数据的版本中的所有记录的主关键字的值均与所述插入操作对应的主关键字的值不同。

应理解，主关键字(Primary Key)通常是表中的一个或多个字段，它的值用于唯一标识表中的某一条记录。主关键字是一种唯一关键字，表定义的一部分。主关键字的值不可重复，例如，关系数据库管理系统 Microsoft Office Access 在表中添加新记录时，会自动检查该新记录的主关键字的值，不允许该新记录的主关键字的值与表中其他记录的主关键字的值重复。主关键字也不可为空（NULL）。主关键字能够加快数据库的操作速度。

15 因此，在本申请技术方案中，在所述服务器当前存储的所述目标数据的版本中不存在所述插入操作对应的主关键字的情况下，说明所述服务器当前存储的所述目标数据的版本中不存在与所述第一事务的插入操作相同的记录，因此，所述第一事务的数据操作与所述服务器当前存储的所述目标数据的版本不存在数据冲突。

在第一方面的第一种可能的实现方式中，所述服务器在确定所述服务器当前存储的所述目标数据的版本中存在所述插入操作对应的主关键字的情况下，确定所述第一事务的数据操作与所述服务器当前存储的所述目标数据的版本存在数据冲突。

25 应理解，在所述服务器当前存储的所述目标数据的版本中存在所述插入操作的主关键字的情况下，说明所述服务器当前存储的所述目标数据的版本中已经存在与所述第一事务的插入操作相同的记录，因此，所述第一事务的数据操作与所述服务器当前存储的所述目标数据的版本存在数据冲突。

结合第一方面，在第一方面的第二种可能的实现方式中，所述第一事务的数据操作30 包括更新操作或删除操作，所述事务提交请求中还包括所述第一版本的时间戳，所述方法还包括：在所述服务器当前存储的所述目标数据的版本的时间戳与所述第一版本的时间戳一致的情况下，所述服务器确定所述第一事务的数据操作与所述服务器当前存储的所述目标数据的版本不存在数据冲突。

35 每对目标数据做一次修改，生成对应的一个新版本，对应地也会设置一个对应的时间戳。如果目标数据的两个版本的时间戳相同，则说明这两个版本是所述目标数据的同一个版本。

因此，在本申请技术方案中，在所述服务器当前存储的所述目标数据的版本的时间戳与所述第一版本的时间戳一致的情况下，说明所述服务器当前存储的所述目标数据的版本与所述第一版本是同一个版本，换句话说，从服务器向客户端发送所述第一版本开始到服务器接收到所述事务提交请求的期间，服务器中存储的所述目标数据并没有被其

他事务修改，因此，所述第一事务的数据操作（即更新操作或删除操作）与所述服务器当前存储的所述目标数据的版本并没有数据冲突。

在第一方面的第二种可能的实现方式中，所述服务器在确定所述服务器当前存储的所述目标数据的版本的时间戳与所述第一版本的时间戳不一致的情况下，确定所述第一事务的数据操作与所述服务器当前存储的所述目标数据的版本存在数据冲突。

所述服务器当前存储的所述目标数据的版本的时间戳与所述第一版本的时间戳不一致，说明从服务器向客户端发送所述第一版本开始到服务器接收到所述事务提交请求的期间，服务器中存储的上述目标数据已经被其它事务作了修改。在这种情况下，如果按照所述第一事务的数据操作对所述服务器当前存储的所述目标数据的版本做修改，会破坏事务与数据库的一致性，因此，认为在所述服务器当前存储的所述目标数据的版本的时间戳与所述第一版本的时间戳不一致的情况下，所述第一事务的数据操作与所述服务器当前存储的所述目标数据的版本之间存在数据冲突。

结合第一方面或第一方面的第一种或第二种可能的实现方式，在第一方面的第三种可能的实现方式中，在所述服务器获得所述目标数据的提交更新的版本之后，所述方法还包括：所述服务器为所述目标数据的提交更新的版本设置新的时间戳。

还应理解，如果在所述服务器提交所述第一事务之后，所述服务器接收到其他客户端的用于请求所述目标数据的数据库访问请求，则所述服务器向所述其他客户端发送的所述目标数据的第二版本，所述第二版本即为经过所述第一事务的数据操作修改之后的所述目标数据的版本，应理解，所述目标数据的第二版本的时间戳与所述目标数据的第一版本的时间戳不同。

结合第一方面或第一方面的第一种或第二种可能的实现方式，在第一方面的第三种可能的实现方式中，所述方法还包括：在确定所述第一事务的数据操作与所述服务器当前存储的所述目标数据的版本不存在数据冲突的情况下，所述服务器向所述客户端发送用于指示所述第一事务的提交无效的响应。

在上述各个实现方式中，所述服务器可以接收多个客户端的用于请求所述目标数据的数据库访问请求，每个客户端的数据访问请求，向所述每个客户端发送对应的所述目标数据的一个版本。当接收到所述每个客户端发送的事务提交请求时，在确定要提交的事务的数据操作与所述服务器当前存储的所述目标数据的版本不存在数据冲突的情况下，完成对应事务的提交。

具体地，例如，所述服务器在第一时刻 t_1 接收第一客户端的用于请求目标数据的数据库访问请求；所述服务器向所述第一客户端发送所述目标数据的第一版本，所述第一版本为所述服务器当前存储的所述目标数据的版本。所述服务器在第二时刻 t_2 接收第二客户端的用于请求目标数据的数据库访问请求；所述服务器向所述第二客户端发送所述目标数据的第二版本，所述第二版本为所述服务器当前存储的所述目标数据的版本。应理解，所述第一版本与所述第二版本可能相同，也可能不同。假设所述第一时刻 t_1 早于所述第二时刻 t_2 ，如果在所述第一时刻 t_1 到所述第二时刻 t_2 期间，服务器中存储的所述目标数据没有被修改，则所述第一版本与所述第二版本为同一个版本；如果在所述第一时刻 t_1 到所述第二时刻 t_2 期间，服务器中存储的所述目标数据被其他事务修改，则所述第一版本与所述第二版本为不同的版本，即所述第一版本的时间戳与所述第二版本的时间戳不同。当所述服务器接收到所述第一客户端或所述第二客户端的事务提交请求时，

通过判断要提交的事务的数据操作与所述服务器当前存储的所述目标数据的版本是否存在数据冲突，来确定是否提交事务。

第二方面提供了一种处理数据库事务的方法，该方法包括：客户端获取用于请求执行第一事务的事务请求，所述第一事务要基于目标数据进行执行；所述客户端向服务器发送用于请求所述目标数据的数据库访问请求；所述客户端接收所述服务器发送的所述目标数据的第一版本，所述第一版本为所述服务器当前存储的所述目标数据的版本，所述服务器当前存储的所述目标数据的版本为所述目标数据的最近一次提交更新的版本；所述客户端基于所述第一版本执行所述第一事务；在执行完所述第一事务后，所述客户端向所述服务器发送事务提交请求，所述事务提交请求用于请求提交所述第一事务，所述事务提交请求中包括用于记录所述第一事务的数据操作的日志，以使得所述客户端根据所述日志，在确定所述第一事务的数据操作与所述服务器当前存储的所述目标数据的版本不存在数据冲突的情况下，按照所述第一事务的数据操作修改所述服务器当前存储的所述目标数据的版本，获得所述目标数据的提交更新的版本，以完成所述第一事务的提交。

在本申请技术方案中，服务器接收到客户端发送的请求目标数据的数据库访问请求后，通过向客户端发送目标数据的第一版本，该第一版本为服务器当前存储的目标数据的版本，且该版本为目标数据的最近一次提交更新的版本，使得可以在客户端本地缓存中基于该第一版本完成事务的执行；在接收到客户端发送的事务提交请求时，在确定要提交的事务的数据操作与服务器中当前存储的该目标数据的版本不存在数据冲突的情况下，按照要提交的事务的数据操作修改服务器当前存储的该目标数据的版本，获得该目标数据的提交更新的版本，以完成事务的提交。应理解，通过向客户端发送目标数据的第一版本，使得在客户端执行事务，能够保证并发事务之间的隔离性，即实现了并发的读操作之间没有影响，并发的读操作与写操作之间也没有影响，并发的写操作与写操作之间也没有影响。在服务器提交事务，并在确定待提交的事务的数据操作与服务器当前存储的目标数据的版本不存在数据冲突的情况下，才执行事务的提交，能够保证事务与数据库的一致性。此外，通过在客户端基于目标数据的第一版本执行事务，使得服务器上对该目标数据无需维护多个版本，只需维护该目标数据的最近一次提交更新的版本即可，相对于现有的 MVCC 技术，本申请能够有效降低数据库的存储管理复杂度，从而减小服务器的存储负担。因此，本申请提供了一种能够支持多个读操作和/或写操作同时并发的无锁事务机制，并且能够有效降低数据库的存储管理复杂度。

结合第二方面，在第二方面的第一种可能的实现方式中，所述第一事务的数据操作包括插入操作，所述日志中包括所述插入操作对应的主关键字。

具体地，所述主关键字用于所述服务器判断是否所述第一事务的数据操作存在数据冲突，具体方法详见上文描述，这里不再赘述。

结合第二方面，在第二方面的第一种可能的实现方式中，所述第一事务的数据操作包括更新操作或删除操作，所述事务提交请求中还包括所述第一版本的时间戳。

具体地，所述事务提交请求中包括所述第一版本的时间戳，用于所述服务器判断是否所述第一事务的数据操作存在数据冲突，具体方法详见上文描述，这里不再赘述。

第三方面提供了一种服务器，所述服务器用于执行上述第一方面或第一方面的任一方面的可能实现方式中的方法。

具体地，所述服务器可以包括用于执行第一方面或第一方面的任一可能的实现方式中的方法的模块。

第四方面，提供了一种客户端，所述客户端用于执行上述第二方面或第二方面的任一方面的可能实现方式中的方法。

5 具体地，所述客户端可以包括用于执行第二方面或第二方面的任一可能的实现方式中的方法的模块。

第五方面提供了一种服务器，包括：收发器，用于接收客户端发送的数据库访问请求，所述数据库访问请求用于请求目标数据；所述收发器还用于，基于所述数据库访问请求，向所述客户端发送所述目标数据的第一版本，所述第一版本为所述服务器当前存储的所述目标数据的版本，所述服务器只保存所述目标数据的最近一次提交更新的版本；
10 所述收发器还用于，接收所述客户端基于所述第一版本执行完第一事务后发送的事务提交请求，所述事务提交请求用于请求提交所述第一事务，所述事务提交请求中包括用于记录所述第一事务的数据操作的日志；无锁内存存储引擎，用于根据所述日志，在确定所述第一事务的数据操作与所述服务器当前存储的所述目标数据的版本不存在数据冲突的情况下，按照所述第一事务的数据操作修改所述服务器当前存储的所述目标数据的版本，
15 获得所述目标数据的提交更新的版本，以完成所述第一事务的提交。

在本申请技术方案中，服务器接收到客户端发送的请求目标数据的数据库访问请求后，通过向客户端发送目标数据的第一版本，该第一版本为服务器当前存储的目标数据的版本，且该版本为目标数据的最近一次提交更新的版本，使得可以在客户端本地缓存
20 中基于该第一版本完成事务的执行；在接收到客户端发送的事务提交请求时，在确定要提交的事务的数据操作与服务器中当前存储的该目标数据的版本不存在数据冲突的情况下，按照要提交的事务的数据操作修改服务器当前存储的该目标数据的版本，获得该目标数据的提交更新的版本，以完成事务的提交。应理解，通过向客户端发送目标数据的第一版本，使得在客户端执行事务，能够保证并发事务之间的隔离性，即实现了并发的
25 读操作之间没有影响，并发的读操作与写操作之间也没有影响，并发的写操作与写操作之间也没有影响。在服务器提交事务，并在确定待提交的事务的数据操作与服务器当前存储的目标数据的版本不存在数据冲突的情况下，才执行事务的提交，能够保证事务与数据库的一致性。此外，通过在客户端基于目标数据的第一版本执行事务，使得服务器上对该目标数据无需维护多个版本，只需维护该目标数据的最近一次提交更新的版本即可，
30 相对于现有的 MVCC 技术，本申请能够有效降低数据库的存储管理复杂度，从而减小服务器的存储负担。因此，本申请提供了一种能够支持多个读操作和/或写操作同时并发的无锁事务机制，并且能够有效降低数据库的存储管理复杂度。

结合第三方面，在第三方面的第一种可能的实现方式中，所述第一事务的数据操作包括插入操作，所述日志中包括所述插入操作对应的主关键字，所述无锁内存存储引擎
35 还用于，在所述服务器当前存储的所述目标数据的版本中不存在所述主关键字的情况下，确定所述第一事务的数据操作与所述服务器当前存储的所述目标数据的版本不存在数据冲突。

结合第三方面，在第三方面的第二种可能的实现方式中，所述第一事务的数据操作包括更新操作或删除操作，所述事务提交请求中还包括所述第一版本的时间戳，所述无
40 锁内存存储引擎还用于，在所述服务器当前存储的所述目标数据的版本的时间戳与所述

第一版本的时间戳一致的情况下，确定所述第一事务的数据操作与所述服务器当前存储的所述目标数据的版本不存在数据冲突。

结合第三方面或第三方面的第一种或第二种可能的实现方式中，在第三方面的第三种可能的实现方式中，所述无锁内存存储引擎还用于，在所述服务器获得所述目标数据的提交更新的版本之后，为所述目标数据的提交更新的版本设置新的时间戳。

结合第三方面或第三方面的第一种或第二种或第三种可能的实现方式中，在第三方面的第四种可能的实现方式中，所述收发器用于，在确定所述第一事务的数据操作与所述服务器当前存储的所述目标数据的版本不存在数据冲突的情况下，向所述客户端发送用于指示所述第一事务的提交无效的响应。

第四方面提供了一种客户端，所述客户端包括处理器与存储器，所述存储器用于存储指令，所述处理器用于执行所述存储器存储的指令，并且对所述存储器中存储的指令的执行使得所述处理器执行第二方面或第二方面的任一方面的可能实现方式中的方法。

在上述各个实现方式中，所述目标数据可以为一行数据（数据库中的一行数据也称为一个或一条记录），这种情况下，所述目标数据的版本指的是这一行数据的版本。或者，所述目标数据可以为多行数据，这种情况下，所述目标数据的版本指的是这多行数据的版本。或者，所述目标数据直接为一个数据库表，这种情况下，所述目标数据的版本指的是这个数据库表的版本。

在上述某些实现方式中，所述第一事务的数据操作为插入操作指的是，所述第一事务在数据库中插入数据（也可称为插入记录），而该插入的数据原先并不存在于数据库中。所述第一事务的数据操作为删除操作指的是，所述第一事务删除数据库中原本存在的数据（或记录）。所述第一事务的数据操作为更新操作指的是，所述第一事务要修改数据库中原本存在的数据的数值。

基于上述技术方案，本申请提供了一种能够支持多个读操作和/或写操作同时并发的无锁事务机制，并且能够有效降低数据库的存储管理复杂度。

附图说明

图 1 示出了本发明实施例的系统架构的示意图。

图 2 示出了根据本发明实施例提供的处理数据库事务的方法的示意性流程图。

图 3 示出了根据本发明实施例提供的服务器的示意性框图。

图 4 示出了根据本发明实施例提供的客户端的示意性框图。

图 5 示出了根据本发明实施例提供的服务器的另一示意性框图。

图 6 示出了根据本发明实施例提供的服务器的另一示意性框图。

具体实施方式

下面将结合附图，对本申请中的技术方案进行清楚描述。

图 1 示出了本发明实施例的系统架构 100 的示意图。该系统包括客户端（Client）和服务器。服务器包括数据存储系统、无锁内存存储引擎（Lock-Free Memory Storage Engine）和远程过程调用（Remote Procedure Call, RPC）模块。数据存储系统中存储有数据库。无锁内存存储引擎用于支持客户端对数据库的实时高效访问。RPC 模块用于客户端与服务端之间的通信。例如服务器通过 RPC 模块接收客户端发送的数据库访问请求，通过无

锁内存存储引擎为客户端提供对应的数据，再通过 RPC 模块向客户端发送数据。客户端包括应用程序模块（如图 1 所示的应用程序 1、2）、客户端缓存（Client Cache）和 RPC 模块。客户端缓存位于客户端的内存中，使用类似于数组的数据结构存储数据。客户端内的 RPC 模块也用于客户端与服务端之间的通信。客户端缓存用于存储服务器下发的数据。例如，客户端通过应用程序模块获取事务请求，通过 RPC 模块向服务器发送数据库访问请求，然后再通过 RPC 模块接收服务器发送的数据，将接收到的数据缓存到客户端缓存，然后在客户端缓存内执行事务，可以实现事务的隔离。应理解，客户端内还可以包括通用内存数据库核心应用程序编程接口（General Memory DataBase Core Application Programming Interface, GMDDB Core API），具体地，GMDDB Core API 包括查询应用程序编程接口（Query Application Programming Interface, Query API）和关键字值应用程序编程接口（Key Value Application Programming Interface, KV API），其中，Query API 用于支持条件范围查询；Kv API 用于主键值查询。

在本发明实施例中，在服务器中，针对一个数据仅保存该数据的最近一次提交更新的版本，对应地，也仅保存该数据最近一次提交更新的版本的时间戳。该数据的最近一次提交更新的版本指的是该数据的最近一次经过要提交的事务的修改操作之后的版本，即该数据的最近一次提交更新的版本满足事务与数据库的一致性。

本发明实施例中提及的数据的版本均指满足事务与数据库一致性的版本。

图 2 示出了本发明实施例提供的处理数据库事务的方法 100 的流程图，该方法 100 包括：

S110，客户端获取用于请求执行第一事务的事务请求，该第一事务要基于目标数据进行执行。

具体地，该客户端例如为图 1 中所示的客户端。例如，该客户端通过应用程序获取用户的事务请求。

S120，该客户端向服务器发送用于请求该目标数据的数据库访问请求。

具体地，该服务器例如为图 1 中所示的服务器。

应理解，数据库存储于服务器的数据存储系统中（如图 1 所示），客户端要访问数据库中的数据时，需要向服务器发送数据库访问请求。

还应理解，该客户端还为该第一事务分配缓存。

S130，服务器接收客户端发送的数据库访问请求，并基于该数据库访问请求，向该客户端发送该目标数据的第一版本，该第一版本为该服务器当前存储的该目标数据的版本，该服务器当前存储的该目标数据的版本为该目标数据的最近一次提交更新的版本。

在本发明实施例中，该服务器中只保存该目标数据的最近一次提交更新的版本，并不保存该目标数据的历史版本，相比于现有的 MVCC 技术，能够简化数据库的存储复杂性，也能减轻服务器的存储负担。

S140，该客户端接收该服务器发送的该目标数据的第一版本，并将将该第一版本存储到该客户端的缓存，并在该客户端的缓存内基于该第一版本执行该第一事务。

在本发明实施例中，是在客户端内执行事务，而非在服务器中执行，这样也使得服务器无需维护该目标数据的多个版本。

S150，在执行完该第一事务后，该客户端向该服务器发送事务提交请求，该事务提交请求用于请求提交该第一事务，该事务提交请求中包括用于记录该第一事务的数据操

作的日志。

应理解，该日志用于记录该第一事务的数据操作，具体地，例如该第一事务的数据操作为插入操作、删除操作或更新操作。其中，插入操作指的是，该第一事务在该目标数据中插入一行数据（也可称为插入一条记录），而该插入的一行数据原先并不存在于该目标数据中。删除操作指的是，该第一事务删除该目标数据中的一行数据或多行数据，或可描述为删除该目标数据中的一条或多条记录。更新操作指的是，该第一事务修改该目标数据中的某些数据的数值。

本发明实施例中提到的目标数据，指的是第一事务所依赖的数据资源，即该第一事务要对该目标数据做一系列的数据操作。具体地，该目标数据可以是数据库中的一条或多条记录。应理解，在数据库中，一行数据称为一个或一条记录，它表达有一定意义的信息组合。每条记录通常都有一个主关键字（Primary Key），用于唯一指示该条记录。

S160，该服务器接收该客户端基于该第一版本执行完该第一事务后发送的事务提交请求，并根据该日志记录，判断第一事务的数据操作与该服务器当前存储的该目标数据的版本是否存在数据冲突，若是，转到 S180，若否，转到 S170。

具体地，如果按照第一事务对该服务器当前存储的该目标数据的版本作修改会破坏数据库的一致性，则认为该第一事务的数据操作与该服务器当前存储的该目标数据的版本之间存在数据冲突，反之不存在冲突。

S170，该服务器在确定该第一事务的数据操作与该服务器当前存储的该目标数据的版本不存在数据冲突的情况下，按照该第一事务的数据操作修改该服务器当前存储的该目标数据的版本，获得该目标数据的提交更新的版本，以完成该第一事务的提交。

在该服务器获得该目标数据的提交更新的版本之后，该服务器为该目标数据的提交更新的版本设置新的时间戳。

具体地，在 S170 中，该服务器可以向客户端发送用于指示该第一事务提交成功的响应。对应地，客户端也可以向用户呈现用于指示第一事务成功提交的响应。

S180，该服务器在确定该第一事务的数据操作与该服务器当前存储的该目标数据的版本存在数据冲突的情况下，不对该服务器当前存储的该目标数据的版本做任何修改操作。

具体地，在 S180 中，该服务器向该客户端发送用于指示该第一事务的提交无效的响应。

在本发明实施例中，服务器接收到客户端发送的请求目标数据的数据库访问请求后，通过向客户端发送目标数据的第一版本，该第一版本为服务器当前存储的目标数据的版本，且该版本为目标数据的最近一次提交更新的版本，使得可以在客户端本地缓存中基于该第一版本完成事务的执行；在接收到客户端发送的事务提交请求时，在确定要提交的事务的数据操作与服务器中当前存储的该目标数据的版本不存在数据冲突的情况下，按照要提交的事务的数据操作修改服务器当前存储的该目标数据的版本，获得该目标数据的提交更新的版本，以完成事务的提交。应理解，通过向客户端发送目标数据的第一版本，使得在客户端执行事务，能够保证并发事务之间的隔离性，即实现了并发的读操作之间没有影响，并发的读操作与写操作之间也没有影响，并发的写操作与写操作之间也没有影响。在服务器提交事务，并在确定待提交的事务的数据操作与服务器当前存储的目标数据的版本不存在数据冲突的情况下，才执行事务的提交，能够保证事务与数据

库的一致性。此外，通过在客户端基于目标数据的第一版本执行事务，使得服务器上对该目标数据无需维护多个版本，只需维护该目标数据的最近一次提交更新的版本即可，相对于现有的 MVCC 技术，本申请能够有效降低数据库的存储管理复杂度，从而减小服务器的存储负担。因此，本申请提供了一种能够支持多个读操作和/或写操作同时并发的无锁事务机制，并且能够有效降低数据库的存储管理复杂度。

在客户端缓存内执行的第一事务的数据操作可以为插入操作、删除操作或更新操作，针对第一事务的数据操作的不同，服务器判断该第一事务的数据操作与该服务器当前存储的该目标数据的版本是否存在数据冲突的方法也不同。

可选地，作为一个实施例，在 S140 中，该客户端将服务器下发的第一版本存储到客户端缓存，并在客户端缓存内基于该第一版本执行该第一事务的数据操作，该第一事务的数据操作为插入操作。在 S150 中，该客户端向该服务器发送事务提交请求中包括该第一事务的日志，该日志用于记录该第一事务的数据操作，即插入操作，并且该日志中包括该插入操作的主关键字（Primary Key）。在 S160 中，服务器根据该日志，判断第一事务的数据操作与该服务器当前存储的该目标数据的版本是否存在数据冲突，包括：该服务器根据日志获知该第一事务的数据操作为插入操作，并从日志中获知该插入操作的主关键字，判断该服务器当前存储的该目标数据的版本中是否存在该插入操作的主关键字，若否，确定该第一事务的数据操作与该服务器当前存储的该目标数据的版本不存在数据冲突；若是，确定该第一事务的数据操作与该服务器当前存储的该目标数据的版本存在数据冲突。

具体地，该第一事务的数据操作为插入操作的是，该第一事务要在该目标数据中插入一条或多条记录，而插入的记录原先并不存在于该目标数据的第一版本中。

应理解，主关键字(Primary Key)通常是表中的一个或多个字段，它的值用于唯一标识表中的某一条记录。主关键字是一种唯一关键字，表定义的一部分。主关键字的值不可重复，例如，关系数据库管理系统 Microsoft Office Access 在表中添加新记录时，会自动检查该新记录的主关键字的值，不允许该新记录的主关键字的值与表中其他记录的主关键字的值重复。主关键字也不可为空（NULL）。主关键字能够加快数据库的操作速度。

也就是说，第一事务的日志中包括该插入操作的主关键字能够唯一指示该插入操作。

应理解，在该服务器当前存储的该目标数据的版本中不存在该插入操作的主关键字的情况下，说明该服务器当前存储的该目标数据的版本中不存在与该第一事务的插入操作相同的记录，因此，该第一事务的数据操作与该服务器当前存储的该目标数据的版本不存在数据冲突。

还应理解，在该服务器当前存储的该目标数据的版本中存在该插入操作的主关键字的情况下，说明该服务器当前存储的该目标数据的版本中已经存在与该第一事务的插入操作相同的记录，因此，该第一事务的数据操作与该服务器当前存储的该目标数据的版本存在数据冲突。

可选地，作为一个实施例，在 S140 中，该客户端将服务器下发的第一版本存储到客户端缓存，并在客户端缓存内基于该第一版本执行该第一事务的数据操作，该第一事务的数据操作为更新操作或删除操作。在 S150 中，该客户端向该服务器发送事务提交请求中包括该第一事务的日志，该日志用于记录该第一事务的数据操作，即更新操作或删除操作，该事务提交请求中还包括该第一版本的时间戳。在 S160 中，服务器根据该日志记

录,判断第一事务的数据操作与该服务器当前存储的该目标数据的版本是否存在数据冲突,包括:该服务器根据日志获知该第一事务的数据操作为更新操作或删除操作,并从事务提交请求中获知执行该第一事务所基于的第一版本的时间戳,该服务器判断该服务器当前存储的该目标数据的版本的时间戳是否与该第一版本的时间戳一致,若是,确定该第一事务的数据操作与该服务器当前存储的该目标数据的版本不存在数据冲突;若否,确定该第一事务的数据操作与该服务器当前存储的该目标数据的版本存在数据冲突。

具体地,当该第一事务的数据操作为更新操作时,即该第一事务要修改该目标数据的第一版本中原本存在的数据(或记录)的数值。当该第一事务的数据操作为删除操作时,即该第一事务要删除该目标数据的第一版本中原本存在的数据(或记录)。

应理解,版本的时间戳用于指示该版本被更新的时间。如果该目标数据的版本的时间戳相同,则说明这两个版本是该目标数据的同一个版本,否则,为两个不同的版本。

还应理解,在该服务器当前存储的该目标数据的版本的时间戳与该第一版本的时间戳一致的情况下,说明从服务器向客户端发送该第一版本开始到服务器接收到该事务提交请求的期间,服务器中存储的该目标数据并没有被其他事务所修改,因此,该第一事务的数据操作与该服务器当前存储的该目标数据的版本不存在数据冲突。

还应理解,在该服务器当前存储的该目标数据的版本的时间戳与该第一版本的时间戳不一致的情况下,说明从服务器向客户端发送该第一版本开始到服务器接收到该事务提交请求的期间,服务器中存储的该目标数据被其他事务所修改,因此,该第一事务的数据操作与该服务器当前存储的该目标数据的版本存在数据冲突。

因此,在本发明实施例中,服务器接收到客户端发送的请求目标数据的数据库访问请求后,通过向客户端发送目标数据的第一版本,该第一版本为服务器当前存储的目标数据的版本,且该版本为目标数据的最近一次提交更新的版本,使得可以在客户端本地缓存中基于该第一版本完成事务的执行;在接收到客户端发送的事务提交请求时,在确定要提交的事务的数据操作与服务器中当前存储的该目标数据的版本不存在数据冲突的情况下,按照要提交的事务的数据操作修改服务器当前存储的该目标数据的版本,获得该目标数据的提交更新的版本,以完成事务的提交。应理解,通过向客户端发送目标数据的第一版本,使得在客户端执行事务,能够保证并发事务之间的隔离性,即实现了并发的读操作之间没有影响,并发的读操作与写操作之间也没有影响,并发的写操作与写操作之间也没有影响。在服务器提交事务,并在确定待提交的事务的数据操作与服务器当前存储的目标数据的版本不存在数据冲突的情况下,才执行事务的提交,能够保证事务与数据库的一致性。此外,通过在客户端基于目标数据的第一版本执行事务,使得服务器上对该目标数据无需维护多个版本,只需维护该目标数据的最近一次提交更新的版本即可,相对于现有的MVCC技术,本申请能够有效降低数据库的存储管理复杂度,从而减小服务器的存储负担。因此,本申请提供了一种能够支持多个读操作和/或写操作同时并发的无锁事务机制,并且能够有效降低数据库的存储管理复杂度。

图3示出了根据本发明实施例提供的服务器200的示意性框图,该服务器200包括:

接收模块210,用于接收客户端发送的数据库访问请求,该数据库访问请求用于请求目标数据;

发送模块220,用于基于该接收模块接收的该数据库访问请求,向该客户端发送该目标数据的第一版本,该第一版本为该服务器当前存储的该目标数据的版本,该服务器当

前存储的该目标数据的版本为该目标数据的最近一次提交更新的版本;

该接收模块 210 还用于, 接收该客户端基于该第一版本执行完第一事务后发送的事务提交请求, 该事务提交请求用于请求提交该第一事务, 该事务提交请求中包括用于记录该第一事务的数据操作的日志;

- 5 处理模块 230, 用于根据该接收模块接收的该日志, 在确定该第一事务的数据操作与该服务器当前存储的该目标数据的版本不存在数据冲突的情况下, 按照该第一事务的数据操作修改该服务器当前存储的该目标数据的版本, 获得该目标数据的提交更新的版本, 以完成该第一事务的提交。

- 10 具体地, 接收模块 210 可以由服务器 200 中的接收器或接收器相关组件实现。发送模块 220 可以由服务器 200 中的发送器或发送器相关组件实现。具体地, 接收模块 210 与发送模块 220 可以是两个独立的器件, 也可以由一个具有收发功能的器件实现, 例如, 接收模块 210 与发送模块 220 可以由图 1 中所示的服务器内的远程过程调用模块实现。处理模块 230 可以由服务器 200 中的处理器或处理器相关组件实现。

- 15 因此, 在本发明实施例中, 服务器接收到客户端发送的请求目标数据的数据库访问请求后, 通过向客户端发送目标数据的第一版本, 该第一版本为服务器当前存储的目标数据的版本, 且该版本为目标数据的最近一次提交更新的版本, 使得可以在客户端本地缓存中基于该第一版本完成事务的执行; 在接收到客户端发送的事务提交请求时, 在确定要提交的事务的数据操作与服务器中当前存储的该目标数据的版本不存在数据冲突的情况下, 按照要提交的事务的数据操作修改服务器当前存储的该目标数据的版本, 获得
20 该目标数据的提交更新的版本, 以完成事务的提交。应理解, 通过向客户端发送目标数据的第一版本, 使得在客户端执行事务, 能够保证并发事务之间的隔离性, 即实现了并发的读操作之间没有影响, 并发的读操作与写操作之间也没有影响, 并发的写操作与写操作之间也没有影响。在服务器提交事务, 并在确定待提交的事务的数据操作与服务器当前存储的目标数据的版本不存在数据冲突的情况下, 才执行事务的提交, 能够保证事务与数据库的一致性。此外, 通过在客户端基于目标数据的第一版本执行事务, 使得服
25 务器上对该目标数据无需维护多个版本, 只需维护该目标数据的最近一次提交更新的版本即可, 相对于现有的 MVCC 技术, 本申请能够有效降低数据库的存储管理复杂度, 从而减小服务器的存储负担。因此, 本申请提供了一种能够支持多个读操作和/或写操作同时并发的无锁事务机制, 并且能够有效降低数据库的存储管理复杂度。

- 30 可选地, 作为一个实施例, 该第一事务的数据操作包括插入操作, 该日记中包括该插入操作对应的主关键字, 该处理模块 230 还用于, 在该服务器当前存储的该目标数据的版本中不存在该主关键字的情况下, 确定该第一事务的数据操作与该服务器当前存储的该目标数据的版本不存在数据冲突。

- 35 可选地, 作为一个实施例, 该第一事务的数据操作包括更新操作或删除操作, 该事务提交请求中还包括该第一版本的时间戳, 该处理模块 230 还用于, 在该服务器当前存储的该目标数据的版本的时间戳与该第一版本的时间戳一致的情况下, 确定该第一事务的数据操作与该服务器当前存储的该目标数据的版本不存在数据冲突。

可选地, 作为一个实施例, 该处理模块 230 还用于, 在获得该目标数据的提交更新的版本之后, 为该目标数据的提交更新的版本设置新的时间戳。

- 40 可选地, 作为一个实施例, 该发送模块 220 还用于, 在确定该第一事务的数据操作

与该服务器当前存储的该目标数据的版本不存在数据冲突的情况下，向该客户端发送用于指示该第一事务的提交无效的响应。

应理解，根据本发明实施例的服务器 200 可对应于本发明实施例的处理数据库事务的方法中的服务器，并且服务器 200 中的各个模块的上述和其它操作和/或功能分别为了

5 实现图 2 中的方法的相应流程，为了简洁，在此不再赘述。

图 4 示出了根据本发明实施例提供的客户端 300 的示意性框图，该客户端 300 包括：处理模块 310，用于获取用于请求执行第一事务的事务请求，该第一事务要基于目标数据进行执行；

发送模块 320，用于向服务器发送用于请求该目标数据的数据库访问请求；

10 接收模块 330，用于接收该服务器发送的该目标数据的第一版本，该第一版本为该服务器当前存储的该目标数据的版本，该服务器当前存储的该目标数据的版本为该目标数据的最近一次提交更新的版本；

该处理模块 310 还用于，基于该接收模块接收的该第一版本执行该第一事务；

该发送模块还用于，在该处理模块执行完该第一事务后，向该服务器发送事务提交请求，该事务提交请求用于请求提交该第一事务，该事务提交请求中包括用于记录该第一事务的数据操作的日志，以使得该客户端根据该日志，在确定该第一事务的数据操作与该服务器当前存储的该目标数据的版本不存在数据冲突的情况下，按照该第一事务的数据操作修改该服务器当前存储的该目标数据的版本，获得该目标数据的提交更新的版本，以完成该第一事务的提交。

15

20 具体地，处理模块 310 可以由客户端 300 中的处理器或处理器相关组件实现。发送模块 320 可以由客户端 300 中的发送器或发送器相关组件实现。接收模块 330 可以由客户端 300 中的接收器或接收器相关组件实现。具体地，接收模块 320 与发送模块 330 可以是两个独立的器件，也可以由一个具有收发功能的器件实现，例如，接收模块 320 与发送模块 330 可以由图 1 中所示的客户端内的远程过程调用模块实现。

25 因此，在本发明实施例中，服务器接收到客户端发送的请求目标数据的数据库访问请求后，通过向客户端发送目标数据的第一版本，该第一版本为服务器当前存储的目标数据的版本，且该版本为目标数据的最近一次提交更新的版本，使得可以在客户端本地缓存中基于该第一版本完成事务的执行；在接收到客户端发送的事务提交请求时，在确定要提交的事务的数据操作与服务器中当前存储的该目标数据的版本不存在数据冲突的情况下，按照要提交的事务的数据操作修改服务器当前存储的该目标数据的版本，获得该目标数据的提交更新的版本，以完成事务的提交。应理解，通过向客户端发送目标数据的第一版本，使得在客户端执行事务，能够保证并发事务之间的隔离性，即实现了并发的读操作之间没有影响，并发的读操作与写操作之间也没有影响，并发的写操作与写操作之间也没有影响。在服务器提交事务，并在确定待提交的事务的数据操作与服务器

30 当前存储的目标数据的版本不存在数据冲突的情况下，才执行事务的提交，能够保证事务与数据库的一致性。此外，通过在客户端基于目标数据的第一版本执行事务，使得服务器上对该目标数据无需维护多个版本，只需维护该目标数据的最近一次提交更新的版本即可，相对于现有的 MVCC 技术，本申请能够有效降低数据库的存储管理复杂度，从而减小服务器的存储负担。因此，本申请提供了一种能够支持多个读操作和/或写操作同时并发的无锁事务机制，并且能够有效降低数据库的存储管理复杂度。

35

40

可选地，作为一个实施例，该第一事务的数据操作包括插入操作，该日记中包括该插入操作对应的主关键字。

可选地，作为一个实施例，该第一事务的数据操作包括更新操作或删除操作，该事务提交请求中还包括该第一版本的时间戳。

- 5 应理解，根据本发明实施例的客户端 300 可对应于本发明实施例的处理数据库事务的方法中的客户端，并且客户端 300 中的各个模块的上述和其它操作和/或功能分别为了实现图 2 中的方法的相应流程，为了简洁，在此不再赘述。

图 5 示出了根据本发明实施例提供的服务器 400 的另一示意性框图，该服务器 400 包括：

- 10 收发器 410，用于接收客户端发送的数据库访问请求，所述数据库访问请求用于请求目标数据；

所述收发器 410 还用于，基于所述数据库访问请求，向所述客户端发送所述目标数据的第一版本，所述第一版本为所述服务器当前存储的所述目标数据的版本，所述服务器当前存储的所述目标数据的版本为所述目标数据的最近一次提交更新的版本；

- 15 所述收发器 410 还用于，接收所述客户端基于所述第一版本执行完第一事务后发送的事务提交请求，所述事务提交请求用于请求提交所述第一事务，所述事务提交请求中包括用于记录所述第一事务的数据操作的日志；

- 20 无锁内存存储引擎 420，用于根据所述日志，在确定所述第一事务的数据操作与所述服务器当前存储的所述目标数据的版本不存在数据冲突的情况下，按照所述第一事务的数据操作修改所述服务器当前存储的所述目标数据的版本，获得所述目标数据的提交更新的版本，以完成所述第一事务的提交。

具体地，该收发器 410 例如为图 1 中所示的服务器内的远程过程调用模块。无锁内存存储引擎 420 例如为图 1 中所示的服务器内的无锁内存存储引擎。

- 25 在本发明实施例中，无锁内存存储引擎 420 是一种无锁的内存 (Memory) 存储引擎，能够实现事务对数据库的更新无需事务锁，从而能够支持客户端对数据的实时高效访问。

具体地，无锁内存存储引擎 420 例如为存储于该服务器 400 内的数据库管理系统 (database management system, DBMS)，用于科学地组织和存储数据(即数据库中的数据)。

- 30 因此，在本发明实施例中，服务器接收到客户端发送的请求目标数据的数据库访问请求后，通过向客户端发送目标数据的第一版本，该第一版本为服务器当前存储的目标数据的版本，且该版本为目标数据的最近一次提交更新的版本，使得可以在客户端本地缓存中基于该第一版本完成事务的执行；在接收到客户端发送的事务提交请求时，在确定要提交的事务的数据操作与服务器中当前存储的该目标数据的版本不存在数据冲突的情况下，按照要提交的事务的数据操作修改服务器当前存储的该目标数据的版本，获得
35 该目标数据的提交更新的版本，以完成事务的提交。应理解，通过向客户端发送目标数据的第一版本，使得在客户端执行事务，能够保证并发事务之间的隔离性，即实现了并发的读操作之间没有影响，并发的读操作与写操作之间也没有影响，并发的写操作与写操作之间也没有影响。在服务器提交事务，并在确定待提交的事务的数据操作与服务器当前存储的目标数据的版本不存在数据冲突的情况下，才执行事务的提交，能够保证事务与数据库的一致性。此外，通过在客户端基于目标数据的第一版本执行事务，使得服
40

务器上对该目标数据无需维护多个版本，只需维护该目标数据的最近一次提交更新的版本即可，相对于现有的 MVCC 技术，本申请能够有效降低数据库的存储管理复杂度，从而减小服务器的存储负担。因此，本申请提供了一种能够支持多个读操作和/或写操作同时并发的无锁事务机制，并且能够有效降低数据库的存储管理复杂度。

5 可选地，在本发明实施例中，该无锁内存存储引擎 420 存储于服务器 400 内的一种计算机可读存储介质中，在该计算机可读存储介质上存储有指令，该服务器 400 内的处理器 430 用于执行该计算机可读存储介质上存储的指令，并且对该计算机可读存储介质上存储的指令的执行使得该处理器 430 能够控制该收发器 410 接收信号或者发送信号，还能够处理该收发器 410 接收到的信号，即处理器 430 对该计算机可读存储介质上存储的指令的执行使得：该收发器 410 用于，接收客户端发送的数据库访问请求，该数据库访问请求用于请求目标数据；基于该数据库访问请求，向该客户端发送该目标数据的第一版本，该第一版本为该服务器当前存储的该目标数据的版本，该服务器当前存储的该目标数据的版本为该目标数据的最近一次提交更新的版本；接收该客户端基于该第一版本执行完第一事务后发送的事务提交请求，该事务提交请求用于请求提交该第一事务，15 该事务提交请求中包括用于记录该第一事务的数据操作的日志；该处理器 410 用于，根据该日志，在确定该第一事务的数据操作与该服务器当前存储的该目标数据的版本不存在数据冲突的情况下，按照该第一事务的数据操作修改该服务器当前存储的该目标数据的版本，获得该目标数据的提交更新的版本，以完成该第一事务的提交。

20 可选地，作为一个实施例，该第一事务的数据操作包括插入操作，该日记中包括该插入操作对应的主关键字，该无锁内存存储引擎 420 还用于，在该服务器当前存储的该目标数据的版本中不存在该主关键字的情况下，确定该第一事务的数据操作与该服务器当前存储的该目标数据的版本不存在数据冲突。

25 可选地，作为一个实施例，该第一事务的数据操作包括更新操作或删除操作，该事务提交请求中还包括该第一版本的时间戳，该无锁内存存储引擎 420 还用于，在该服务器当前存储的该目标数据的版本的时间戳与该第一版本的时间戳一致的情况下，确定该第一事务的数据操作与该服务器当前存储的该目标数据的版本不存在数据冲突。

可选地，作为一个实施例，该无锁内存存储引擎 420 还用于，在获得该目标数据的提交更新的版本之后，为该目标数据的提交更新的版本设置新的时间戳。

30 可选地，作为一个实施例，该无锁内存存储引擎 420 还用于，在确定该第一事务的数据操作与该服务器当前存储的该目标数据的版本存在数据冲突的情况下，控制收发器 410 向该客户端发送用于指示该第一事务的提交无效的响应。

35 应理解，在本发明实施例中，该处理器 430 可以是中央处理单元（Central Processing Unit，简称为“CPU”），该处理器 430 还可以是其他通用处理器、数字信号处理器（DSP）、专用集成电路（ASIC）、现成可编程门阵列（FPGA）或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件等。通用处理器可以是微处理器或者该处理器也可以是任何常规的处理器等。

40 在实现过程中，上述方法的各步骤可以通过处理器 430 中的硬件的集成逻辑电路或者软件形式的指令完成。结合本发明实施例所公开的方法的步骤可以直接体现为硬件处理器执行完成，或者用处理器中的硬件及软件模块组合执行完成。软件模块可以位于随机存储器，闪存、只读存储器，可编程只读存储器或者电可擦写可编程存储器、寄存器

等本领域成熟的存储介质中。该存储介质位于存储器，处理器 430 读取存储器中的信息，结合其硬件完成上述方法的步骤。为避免重复，这里不再详细描述。

应理解，根据本发明实施例的服务器 400 可对应于本发明实施例的处理数据库事务的方法中的服务器，以及可以对应于根据本发明实施例的服务器 200，并且服务器 400 中的各个模块的上述和其它操作和/或功能分别为了实现图 2 中的方法的相应流程，为了简洁，在此不再赘述。

图 6 示出了根据本发明实施例提供的客户端 500 的另一示意性框图。该客户端 500 包括处理器 510，存储器 520，系统总线 530，接收器 540 和发送器 550，其中，处理器 510、存储器 520、接收器 540 和发送器 550 通过总线系统 530 相连，该存储器 520 用于存储指令，该处理器 510 用于执行该存储器 520 存储的指令，以控制接收器 540 接收信号，并控制发送器 550 发送信号。其中，该处理器 510 用于获取用于请求执行第一事务的事务请求，该第一事务要基于目标数据进行执行；发送器 550 用于，向服务器发送用于请求该目标数据的数据库访问请求；接收器 540 用于，接收该服务器发送的该目标数据的第一版本，该第一版本为该服务器当前存储的该目标数据的版本，该服务器当前存储的该目标数据的版本为该目标数据的最近一次提交更新的版本；该处理器 510 还用于，将该第一版本存储客户端 500 的缓存，并基于该第一版本执行该第一事务；发送器 550 还用于，在执行完该第一事务后，向该服务器发送事务提交请求，该事务提交请求用于请求提交该第一事务，该事务提交请求中包括用于记录该第一事务的数据操作的日志，以使得该客户端根据该日志，在确定该第一事务的数据操作与该服务器当前存储的该目标数据的版本不存在数据冲突的情况下，按照该第一事务的数据操作修改该服务器当前存储的该目标数据的版本，获得该目标数据的提交更新的版本，以完成该第一事务的提交。

因此，在本发明实施例中，服务器接收到客户端发送的请求目标数据的数据库访问请求后，通过向客户端发送目标数据的第一版本，该第一版本为服务器当前存储的目标数据的版本，且该版本为目标数据的最近一次提交更新的版本，使得可以在客户端本地缓存中基于该第一版本完成事务的执行；在接收到客户端发送的事务提交请求时，在确定要提交的事务的数据操作与服务器中当前存储的该目标数据的版本不存在数据冲突的情况下，按照要提交的事务的数据操作修改服务器当前存储的该目标数据的版本，获得该目标数据的提交更新的版本，以完成事务的提交。应理解，通过向客户端发送目标数据的第一版本，使得在客户端执行事务，能够保证并发事务之间的隔离性，即实现了并发的读操作之间没有影响，并发的读操作与写操作之间也没有影响，并发的写操作与写操作之间也没有影响。在服务器提交事务，并在确定待提交的事务的数据操作与服务器当前存储的目标数据的版本不存在数据冲突的情况下，才执行事务的提交，能够保证事务与数据库的一致性。此外，通过在客户端基于目标数据的第一版本执行事务，使得服务器上对该目标数据无需维护多个版本，只需维护该目标数据的最近一次提交更新的版本即可，相对于现有的 MVCC 技术，本申请能够有效降低数据库的存储管理复杂度，从而减小服务器的存储负担。因此，本申请提供了一种能够支持多个读操作和/或写操作同时并发的无锁事务机制，并且能够有效降低数据库的存储管理复杂度。

可选地，作为一个实施例，该第一事务的数据操作包括插入操作，该日记中包括该插入操作对应的主关键字。

可选地，作为一个实施例，该第一事务的数据操作包括更新操作或删除操作，该事务提交请求中还包括该第一版本的时间戳。

应理解，在本发明实施例中，该处理器 510 可以是中央处理单元（Central Processing Unit，简称为“CPU”），该处理器 510 还可以是其他通用处理器、数字信号处理器（DSP）、
5 专用集成电路（ASIC）、现成可编程门阵列（FPGA）或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件等。通用处理器可以是微处理器或者该处理器也可以是任何常规的处理器等。

该存储器 520 可以包括只读存储器和随机存取存储器，并向处理器 510 提供指令和数据。存储器 520 的一部分还可以包括非易失性随机存取存储器。例如，存储器 520 还
10 可以存储设备类型的信息。

该总线系统 530 除包括数据总线之外，还可以包括电源总线、控制总线和状态信号总线等。但是为了清楚说明起见，在图中将各种总线都标为总线系统 530。

在实现过程中，上述方法的各步骤可以通过处理器 510 中的硬件的集成逻辑电路或者软件形式的指令完成。结合本发明实施例所公开的方法的步骤可以直接体现为硬件处
15 理器执行完成，或者用处理器中的硬件及软件模块组合执行完成。软件模块可以位于随机存储器，闪存、只读存储器，可编程只读存储器或者电可擦写可编程存储器、寄存器等本领域成熟的存储介质中。该存储介质位于存储器 520，处理器 510 读取存储器 520 中的信息，结合其硬件完成上述方法的步骤。为避免重复，这里不再详细描述。

应理解，发送器 550 可以是用于实现发送功能的硬件电路或器件，如天线、网卡等；
20 同样的，接收器 540 也可以是用于实现发送功能的硬件电路或器件，如天线、网卡等，本发明实施例不做限定。

还应理解，接收器 540 与发送器 550 可以由一个具备收发功能的装置实现，例如收发器，具体地，如天线。

应理解，根据本发明实施例的客户端 500 可对应于本发明实施例的处理数据库事务的方法中的客户端，以及可以对应于根据本发明实施例的客户端 300，并且客户端 500
25 中的各个模块的上述和其它操作和/或功能分别为了实现图 2 中的方法的相应流程，为了简洁，在此不再赘述。

应理解，本文中术语“和/或”，仅仅是一种描述关联对象的关联关系，表示可以存在三种关系，例如，A 和/或 B，可以表示：单独存在 A，同时存在 A 和 B，单独存在 B 这
30 三种情况。另外，本文中字符“/”，一般表示前后关联对象是一种“或”的关系。

应理解，在本申请的各种实施例中，上述各过程的序号的大小并不意味着执行顺序的先后，各过程的执行顺序应以其功能和内在逻辑确定，而不应对本发明实施例的实施过程构成任何限定。

本领域普通技术人员可以意识到，结合本文中所公开的实施例描述的各示例的单元
35 及算法步骤，能够以电子硬件、或者计算机软件和电子硬件的结合来实现。这些功能究竟以硬件还是软件方式来执行，取决于技术方案的特定应用和设计约束条件。专业技术人员可以对每个特定的应用来使用不同方法来实现所描述的功能，但是这种实现不应认为超出本申请的范围。

所属领域的技术人员可以清楚地了解到，为描述的方便和简洁，上述描述的系统、
40 装置和单元的具体工作过程，可以参考前述方法实施例中的对应过程，在此不再赘述。

在本申请所提供的几个实施例中，应该理解到，所揭露的系统、装置和方法，可以通过其它的方式实现。例如，以上所描述的装置实施例仅仅是示意性的，例如，所述单元的划分，仅仅为一种逻辑功能划分，实际实现时可以有另外的划分方式，例如多个单元或组件可以结合或者可以集成到另一个系统，或一些特征可以忽略，或不执行。另一点，所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口，装置或单元的间接耦合或通信连接，可以是电性，机械或其它的形式。

所述作为分离部件说明的单元可以是或者也可以不是物理上分开的，作为单元显示的部件可以是或者也可以不是物理单元，即可以位于一个地方，或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部单元来实现本实施例方案的目的。

另外，在本申请各个实施例中的各功能单元可以集成在一个处理单元中，也可以是各个单元单独物理存在，也可以两个或两个以上单元集成在一个单元中。

所述功能如果以软件功能单元的形式实现并作为独立的产品销售或使用，可以存储在一个计算机可读取存储介质中。基于这样的理解，本申请的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的部分可以以软件产品的形式体现出来，该计算机软件产品存储在一个存储介质中，包括若干指令用以使得一台计算机设备（可以是个人计算机，服务器，或者网络设备等）执行本申请各个实施例所述方法的全部或部分步骤。而前述的存储介质包括：U 盘、移动硬盘、只读存储器（Read-Only Memory, ROM）、随机存取存储器（Random Access Memory, RAM）、磁碟或者光盘等各种可以存储程序代码的介质。

以上所述，仅为本申请的具体实施方式，但本申请的保护范围并不局限于此，任何熟悉本技术领域的技术人员在本申请揭露的技术范围内，可轻易想到变化或替换，都应涵盖在本申请的保护范围之内。因此，本申请的保护范围应以所述权利要求的保护范围为准。

权利要求

1、一种处理数据库事务的方法，其特征在于，包括：

服务器接收客户端发送的数据库访问请求，所述数据库访问请求用于请求目标数据；

所述服务器基于所述数据库访问请求，向所述客户端发送所述目标数据的第一版本，

5 所述第一版本为所述服务器当前存储的所述目标数据的版本，所述服务器中只保存所述目标数据的最近一次提交更新的版本；

所述服务器接收所述客户端基于所述第一版本执行完第一事务后发送的事务提交请求，所述事务提交请求用于请求提交所述第一事务，所述事务提交请求中包括用于记录所述第一事务的数据操作的日志；

10 所述服务器根据所述日志，在确定所述第一事务的数据操作与所述服务器当前存储的所述目标数据的版本不存在数据冲突的情况下，按照所述第一事务的数据操作修改所述服务器当前存储的所述目标数据的版本，获得所述目标数据的提交更新的版本，以完成所述第一事务的提交。

2、根据权利要求1所述的方法，其特征在于，所述第一事务的数据操作包括插入操作，所述日志中包括所述插入操作对应的主关键字，所述方法还包括：

在所述服务器当前存储的所述目标数据的版本中不存在所述主关键字的情况下，所述服务器确定所述第一事务的数据操作与所述服务器当前存储的所述目标数据的版本不存在数据冲突。

3、根据权利要求1所述的方法，其特征在于，所述第一事务的数据操作包括更新操作或删除操作，所述事务提交请求中还包括所述第一版本的时间戳，所述方法还包括：

在所述服务器当前存储的所述目标数据的版本的时间戳与所述第一版本的时间戳一致的情况下，所述服务器确定所述第一事务的数据操作与所述服务器当前存储的所述目标数据的版本不存在数据冲突。

4、根据权利要求1至3中任一项所述的方法，其特征在于，在所述服务器获得所述目标数据的提交更新的版本之后，所述方法还包括：所述服务器为所述目标数据的提交更新的版本设置新的时间戳。

5、根据权利要求1至4中任一项所述的方法，其特征在于，所述方法还包括：在确定所述第一事务的数据操作与所述服务器当前存储的所述目标数据的版本存在数据冲突的情况下，所述服务器向所述客户端发送用于指示所述第一事务的提交无效的响应。

6、一种处理数据库事务的方法，其特征在于，包括：

客户端获取用于请求执行第一事务的事务请求，所述第一事务要基于目标数据进行执行；

所述客户端向服务器发送用于请求所述目标数据的数据库访问请求；

35 所述客户端接收所述服务器发送的所述目标数据的第一版本，所述第一版本为所述服务器当前存储的所述目标数据的版本，所述服务器当前存储的所述目标数据的版本为所述目标数据的最近一次提交更新的版本；

所述客户端基于所述第一版本执行所述第一事务；

40 在执行完所述第一事务后，所述客户端向所述服务器发送事务提交请求，所述事务提交请求用于请求提交所述第一事务，所述事务提交请求中包括用于记录所述第一事务的数据操作的日志，以使得所述客户端根据所述日志，在确定所述第一事务的数据操作

与所述服务器当前存储的所述目标数据的版本不存在数据冲突的情况下，按照所述第一事务的数据操作修改所述服务器当前存储的所述目标数据的版本，获得所述目标数据的提交更新的版本，以完成所述第一事务的提交。

7、根据权利要求6所述的方法，其特征在于，所述第一事务的数据操作包括插入操作，所述日记中包括所述插入操作对应的主关键字。

8、根据权利要求6所述的方法，其特征在于，所述第一事务的数据操作包括更新操作或删除操作，所述事务提交请求中还包括所述第一版本的时间戳。

9、一种服务器，其特征在于，包括：

接收模块，用于接收客户端发送的数据库访问请求，所述数据库访问请求用于请求目标数据；

发送模块，用于基于所述接收模块接收的所述数据库访问请求，向所述客户端发送所述目标数据的第一版本，所述第一版本为所述服务器当前存储的所述目标数据的版本，所述服务器中只保存所述目标数据的版本为所述目标数据的最近一次提交更新的版本；

所述接收模块还用于，接收所述客户端基于所述第一版本执行完第一事务后发送的事务提交请求，所述事务提交请求用于请求提交所述第一事务，所述事务提交请求中包括用于记录所述第一事务的数据操作的日志；

处理模块，用于根据所述接收模块接收的所述日志，在确定所述第一事务的数据操作与所述服务器当前存储的所述目标数据的版本不存在数据冲突的情况下，按照所述第一事务的数据操作修改所述服务器当前存储的所述目标数据的版本，获得所述目标数据的提交更新的版本，以完成所述第一事务的提交。

10、根据权利要求9所述的服务器，其特征在于，所述第一事务的数据操作包括插入操作，所述日记中包括所述插入操作对应的主关键字，所述处理模块还用于，在所述服务器当前存储的所述目标数据的版本中不存在所述主关键字的情况下，确定所述第一事务的数据操作与所述服务器当前存储的所述目标数据的版本不存在数据冲突。

11、根据权利要求9所述的服务器，其特征在于，所述第一事务的数据操作包括更新操作或删除操作，所述事务提交请求中还包括所述第一版本的时间戳，所述处理模块还用于，在所述服务器当前存储的所述目标数据的版本的时间戳与所述第一版本的时间戳一致的情况下，确定所述第一事务的数据操作与所述服务器当前存储的所述目标数据的版本不存在数据冲突。

12、根据权利要求9至11中任一项所述的服务器，其特征在于，所述处理模块还用于，在获得所述目标数据的提交更新的版本之后，为所述目标数据的提交更新的版本设置新的时间戳。

13、根据权利要求9至12中任一项所述的服务器，其特征在于，所述发送模块还用于，在确定所述第一事务的数据操作与所述服务器当前存储的所述目标数据的版本存在数据冲突的情况下，向所述客户端发送用于指示所述第一事务的提交无效的响应。

14、一种客户端，其特征在于，包括：

处理模块，用于获取用于请求执行第一事务的事务请求，所述第一事务要基于目标数据进行执行；

发送模块，用于向服务器发送用于请求所述目标数据的数据库访问请求；

接收模块，用于接收所述服务器发送的所述目标数据的第一版本，所述第一版本为

所述服务器当前存储的所述目标数据的版本，所述服务器当前存储的所述目标数据的版本为所述目标数据的最近一次提交更新的版本；

所述处理模块还用于，基于所述接收模块接收的所述第一版本执行所述第一事务；

5 所述发送模块还用于，在所述处理模块执行完所述第一事务后，向所述服务器发送事务提交请求，所述事务提交请求用于请求提交所述第一事务，所述事务提交请求中包括用于记录所述第一事务的数据操作的日志，以使得所述客户端根据所述日志，在确定所述第一事务的数据操作与所述服务器当前存储的所述目标数据的版本不存在数据冲突的情况下，按照所述第一事务的数据操作修改所述服务器当前存储的所述目标数据的版本，获得所述目标数据的提交更新的版本，以完成所述第一事务的提交。

10 15、根据权利要求 14 所述的客户端，其特征在于，所述第一事务的数据操作包括插入操作，所述日记中包括所述插入操作对应的主关键字。

16、根据权利要求 14 所述的客户端，其特征在于，所述第一事务的数据操作包括更新操作或删除操作，所述事务提交请求中还包括所述第一版本的时间戳。

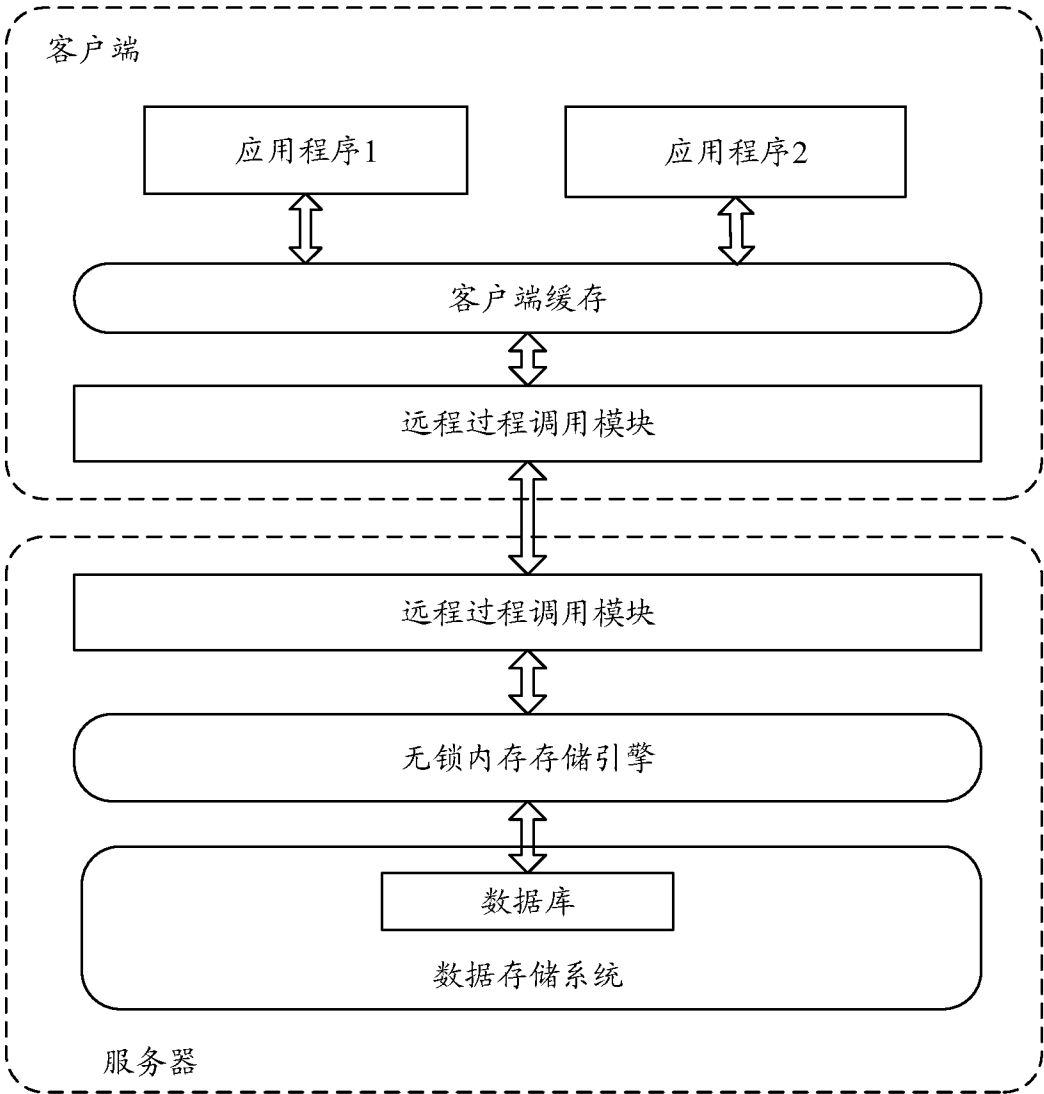


图 1

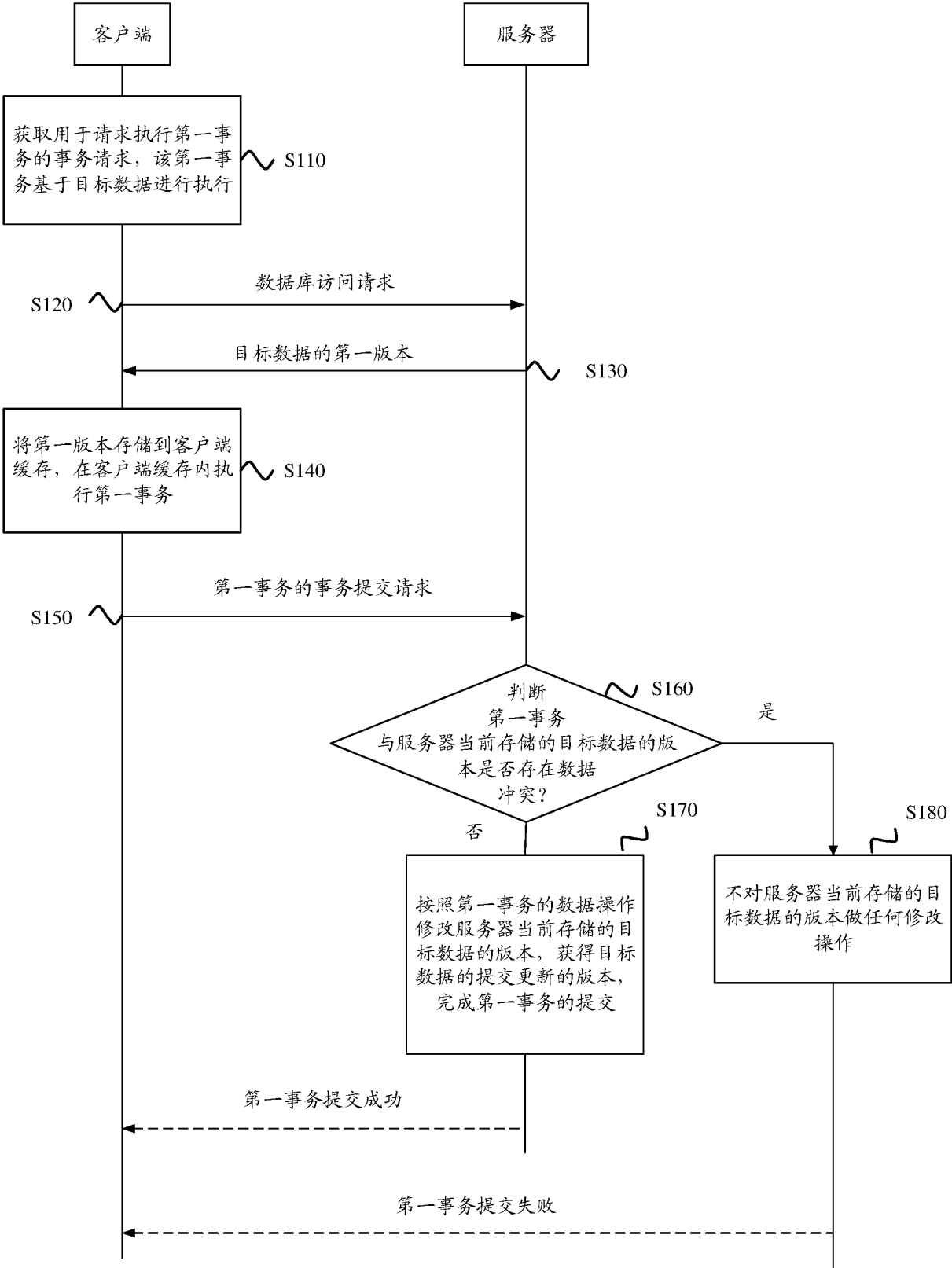


图 2

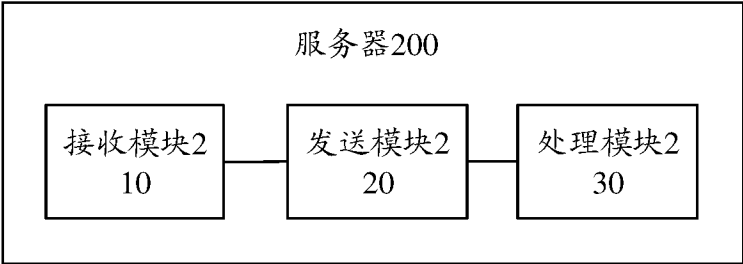


图 3

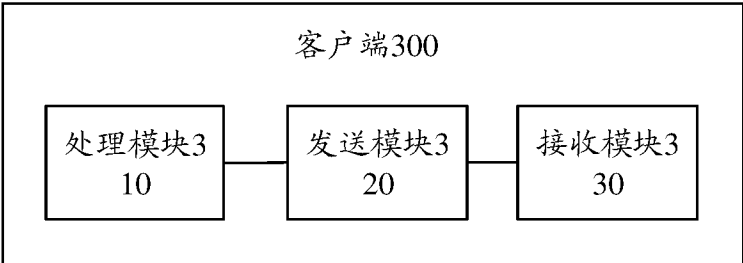


图 4

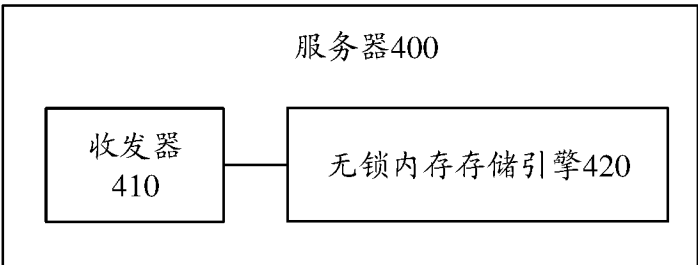


图 5

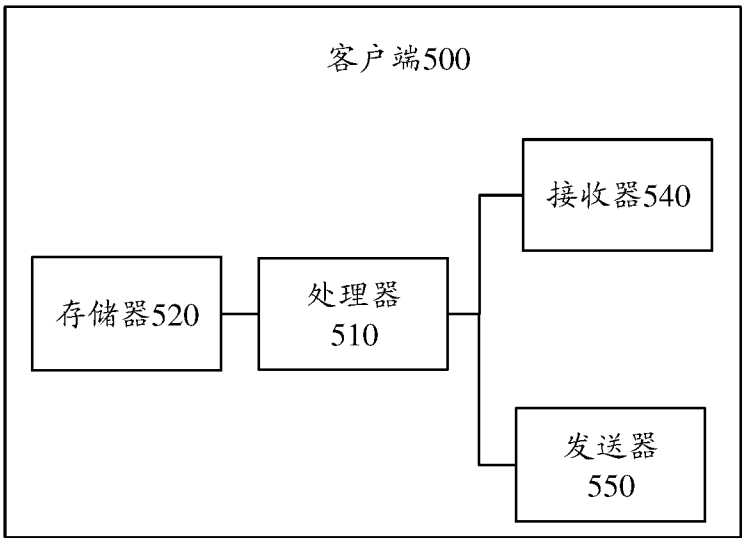


图 6

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2017/089035

A. CLASSIFICATION OF SUBJECT MATTER

G06F 17/30 (2006.01) i; G06F 12/08 (2016.01) i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNPAT; WPI; EPODOC; CNKI; IEEE: affair, parallel, lockless, database, client, server, transaction, execut+, perform+, concurrent, lock+, edition, log, commit+, conflict+, consistency

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	LIU, Lixi, "Research on Transaction Control Mechanism in High Concurrent Environment", ELECTRONIC TECHNOLOGY & INFORMATION SCIENCE, CHINA MASTER'S THESES FULL-TEXT DATABASE, 15 September 2015 (15.09.2015), ISSN: 1674-0246, the main body, pages 9-29	1-16
A	CN 103744936 A (HUAWEI TECHNOLOGIES CO., LTD.), 23 April 2014 (23.04.2014), the whole document	1-16
A	CN 104333512 A (SI-TECH INFORMATION TECHNOLOGY LTD.), 04 February 2015 (04.02.2015), the whole document	1-16
A	CN 105243067 A (MININGLAMP SOFTWARE SYSTEM CO., LTD.), 13 January 2016 (13.01.2016), the whole document	1-16
A	US 2016147906 A1 (SCHRETER, I. et al.), 26 May 2016 (26.05.2016), the whole document	1-16

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents:	"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A" document defining the general state of the art which is not considered to be of particular relevance	"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E" earlier application or patent but published on or after the international filing date	"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&" document member of the same patent family
"O" document referring to an oral disclosure, use, exhibition or other means	
"P" document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search 12 September 2017 (12.09.2017)	Date of mailing of the international search report 20 September 2017 (20.09.2017)
Name and mailing address of the ISA/CN: State Intellectual Property Office of the P. R. China No. 6, Xitucheng Road, Jimenqiao Haidian District, Beijing 100088, China Facsimile No.: (86-10) 62019451	Authorized officer JIN, Xia Telephone No.: (86-10) 61648480

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CN2017/089035

Patent Documents referred in the Report	Publication Date	Patent Family	Publication Date
CN 103744936 A	23 April 2014	US 2016314161 A1	27 October 2016
		EP 3079078 A1	12 October 2016
		WO 2015101025 A1	09 July 2015
CN 104333512 A	04 February 2015	None	
CN 105243067 A	13 January 2016	None	
US 2016147906 A1	26 May 2016	CN 105630863 A	01 June 2016
		EP 3026582 A1	01 June 2016

A. 主题的分类 G06F 17/30(2006.01)i; G06F 12/08(2016.01)i 按照国际专利分类(IPC)或者同时按照国家分类和IPC两种分类		
B. 检索领域 检索的最低限度文献(标明分类系统和分类号) G06F 包含在检索领域中的除最低限度文献以外的检索文献 在国际检索时查阅的电子数据库(数据库的名称, 和使用的检索词(如使用)) CNPAT;WPI;EPDOC;CNKI;IEEE: 数据库, 客户端, 服务器, 事务, 执行, 并行, 并发, 无锁, 版本, 日志, 提交, 冲突, 一致性, database, client, server, transaction, execut+, perform+, concurrent, lock+, edition, log, commit+, conflict+, consistency		
C. 相关文件		
类 型*	引用文件, 必要时, 指明相关段落	相关的权利要求
A	刘笠熙. “高并发环境下的事务控制机制研究” 中国优秀硕士学位论文全文数据库 信息科技辑, 2015年 9月 15日 (2015 - 09 - 15), ISSN: 1674-0246, 正文第9-29页	1-16
A	CN 103744936 A (华为技术有限公司) 2014年 4月 23日 (2014 - 04 - 23) 全文	1-16
A	CN 104333512 A (北京思特奇信息技术股份有限公司) 2015年 2月 4日 (2015 - 02 - 04) 全文	1-16
A	CN 105243067 A (北京明略软件系统有限公司) 2016年 1月 13日 (2016 - 01 - 13) 全文	1-16
A	US 2016147906 A1 (SCHRETER, IVAN 等) 2016年 5月 26日 (2016 - 05 - 26) 全文	1-16
☐ 其余文件在C栏的续页中列出。 ☒ 见同族专利附件。		
* 引用文件的具体类型: “A” 认为不特别相关的表示了现有技术一般状态的文件 “E” 在国际申请日的当天或之后公布的在先申请或专利 “L” 可能对优先权要求构成怀疑的文件, 或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件(如具体说明的) “O” 涉及口头公开、使用、展览或其他方式公开的文件 “P” 公布日先于国际申请日但迟于所要求的优先权日的文件 “T” 在申请日或优先权日之后公布, 与申请不相抵触, 但为了理解发明之理论或原理的在后文件 “X” 特别相关的文件, 单独考虑该文件, 认定要求保护的发明不是新颖的或不具有创造性 “Y” 特别相关的文件, 当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时, 要求保护的发明不具有创造性 “&” 同族专利的文件		
国际检索实际完成的日期 2017年 9月 12日		国际检索报告邮寄日期 2017年 9月 20日
ISA/CN的名称和邮寄地址 中华人民共和国国家知识产权局(ISA/CN) 中国北京市海淀区蓟门桥西土城路6号 100088 传真号 (86-10)62019451		受权官员 金霞 电话号码 (86-10)61648480

国际检索报告
关于同族专利的信息

国际申请号

PCT/CN2017/089035

检索报告引用的专利文件			公布日 (年/月/日)	同族专利			公布日 (年/月/日)
CN	103744936	A	2014年 4月 23日	US	2016314161	A1	2016年 10月 27日
				EP	3079078	A1	2016年 10月 12日
				WO	2015101025	A1	2015年 7月 9日
CN	104333512	A	2015年 2月 4日	无			
CN	105243067	A	2016年 1月 13日	无			
US	2016147906	A1	2016年 5月 26日	CN	105630863	A	2016年 6月 1日
				EP	3026582	A1	2016年 6月 1日