

数据读取方法及装置

本申请要求 2020 年 06 月 19 日递交的申请号为 202010568737.9、发明名称为“数据读取方法及装置”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。

5 技术领域

本说明书实施例涉及数据处理技术领域，特别涉及一种数据读取方法。本说明书一个或者多个实施例同时涉及一种数据读取装置，一种计算设备，以及一种计算机可读存储介质。

10 背景技术

目前市面上的分布式数据库大多不支持备库一致性读，而所谓备库一致性读，是指用同一时间读备库和读主库时读到的结果相同。通常来说，由于主备（即主库和备库）复制存在延迟，备库上的数据状态滞后于主库，导致主备读的结果有可能不同，即备库中的数据与主库中的数据不同步，使得从备库读的数据与主库中的数据不一致。

15 因此需要提供一种可以实现主备读的结果相同，以保证主备库一致性读的数据读取方法。

发明内容

20 鉴于此，本说明书实施例提供了一种数据读取方法。本说明书一个或者多个实施例同时涉及一种数据读取装置，一种计算设备，以及一种计算机可读存储介质，以解决现有技术中存在的技术缺陷。

根据本说明书实施例的第一方面，提供了一种数据读取方法，包括：

接收数据读取事务请求；

基于所述数据读取事务请求获取读取时间戳；

25 在所述读取时间戳小于第一数据库的写入事务的最大提交时间戳的情况下，将携带有所述读取时间戳的数据读取事务请求发送至所述第一数据库进行数据读取。

可选的，所述读取时间戳小于第一数据库的写入事务的最大提交时间戳包括：

所述读取时间戳小于第二数据库针对所述第一数据库的写入事务的最大提交时间戳，其中，所述第一数据库为所述第二数据库的备份数据库。

30 可选的，所述基于所述数据读取事务请求获取读取时间戳之后，还包括：

在所述读取时间戳大于所述第一数据库的写入事务的最大提交时间戳的情况下，在等待预设时长后重新获取所述第一数据库的写入事务的最大提交时间戳。

可选的，所述在等待预设时长后重新获取所述第一数据库的写入事务的最大提交时间戳之后，还包括：

- 5 在所述读取时间戳小于所述第一数据库的写入事务的最大提交时间戳的情况下，获取所述第二数据库的日志序列号；

将携带所述第二数据库的日志序列号的所述数据读取事务请求发送至所述第一数据库，且获取所述第一数据库的日志序列号；

- 10 在确定所述第一数据库的日志序列号与所述第二数据库的日志序列号匹配的情况下，基于所述数据读取事务请求以及所述第一数据库的日志序列号在所述第一数据库读取数据。

可选的，所述获取所述第二数据库的日志序列号包括：

获取所述第二数据库最后一次数据库完整备份的日志序列号；

相应的，所述获取所述第一数据库的日志序列号包括：

- 15 获取所述第一数据库最后一次数据库完整备份的日志序列号。

可选的，所述接收数据读取事务请求之后，还包括：

基于所述数据读取事务请求确定对应的至少一个所述第二数据库。

可选的，所述获取所述第二数据库的日志序列号包括：

获取每个第二数据库的日志序列号；

- 20 相应的，所述将携带所述第二数据库的日志序列号的所述数据读取事务请求发送至所述第一数据库包括：

将携带有所述第二数据库的日志序列号的所述数据读取事务请求发送至与每个第二数据库对应的第一数据库。

可选的，所述数据读取方法应用于分布式数据库。

- 25 根据本说明书实施例的第二方面，提供了一种数据读取装置，包括：

请求接收模块，被配置为接收数据读取事务请求；

时间戳获取模块，被配置为基于所述数据读取事务请求获取读取时间戳；

- 30 数据读取模块，被配置为在所述读取时间戳小于第一数据库的写入事务的最大提交时间戳的情况下，将携带有所述读取时间戳的数据读取事务请求发送至所述第一数据库进行数据读取。

可选的，所述数据读取模块，进一步被配置为：

所述读取时间戳小于第二数据库针对所述第一数据库的写入事务的最大提交时间戳，其中，所述第一数据库为所述第二数据库的备份数据库。

可选的，所述装置，还包括：

- 5 重试模块，被配置为在所述读取时间戳大于所述第一数据库的写入事务的最大提交时间戳的情况下，在等待预设时长后重新获取所述第一数据库的写入事务的最大提交时间戳。

可选的，所述装置，还包括：

- 10 第一序列号获取模块，被配置为在所述读取时间戳小于所述第一数据库的写入事务的最大提交时间戳的情况下，获取所述第二数据库的日志序列号；

第二序列号获取模块，被配置为将携带所述第二数据库的日志序列号的所述数据读取事务请求发送至所述第一数据库，且获取所述第一数据库的日志序列号；

- 15 序列号匹配模块，被配置为在确定所述第一数据库的日志序列号与所述第二数据库的日志序列号匹配的情况下，基于所述数据读取事务请求以及所述第一数据库的日志序列号在所述第一数据库读取数据。

可选的，所述第一序列号获取模块，进一步被配置为：

获取所述第二数据库最后一次数据库完整备份的日志序列号；

相应的，所述第二序列号获取模块，进一步被配置为：

获取所述第一数据库最后一次数据库完整备份的日志序列号。

- 20 可选的，所述装置，还包括：

数据库确定模块，被配置为基于所述数据读取事务请求确定对应的至少一个所述第二数据库。

可选的，所述第一序列号获取模块，进一步被配置为：

获取每个第二数据库的日志序列号；

- 25 相应的，第二序列号获取模块，进一步被配置为：

将携带有所述第二数据库的日志序列号的所述数据读取事务请求发送至与每个第二数据库对应的第一数据库。

可选的，所述数据读取方法应用于分布式数据库。

根据本说明书实施例的第三方面，提供了一种计算设备，包括：

- 30 存储器和处理器；

所述存储器用于存储计算机可执行指令，所述处理器用于执行所述计算机可执行指令，其中，所述处理器执行所述计算机可执行指令时实现所述数据读取方法的步骤。

根据本说明书实施例的第四方面，提供了一种计算机可读存储介质，其存储有计算机可执行指令，该指令被处理器执行时实现所述数据读取方法的步骤。

5 本说明书一个实施例提供了一种数据读取方法及装置，其中，所述数据读取方法包括接收数据读取事务请求；基于所述数据读取事务请求获取读取时间戳；在所述读取时间戳小于第一数据库的写入事务的最大提交时间戳的情况下，将携带有所述读取时间戳的数据读取事务请求发送至所述第一数据库进行数据读取；所述数据读取方法通过获取一次读取时间戳，基于该读取时间戳就可以保证同一数据读取事务请求下读取的分布式
10 备库的数据与对应的主库的数据一致性，极大的提升了用户在分布式备库进行数据的读取及查询体验。

附图说明

图 1 是本说明书一个实施例提供的一种数据读取装置的示意图；
15 图 2 是本说明书一个实施例提供的一种数据读取方法的流程图；
图 3 是本说明书一个实施例提供的第一种数据读取方法的处理过程流程图；
图 4 是本说明书一个实施例提供的第二种数据读取方法的处理过程流程图；
图 5 是本说明书一个实施例提供的一种数据读取装置的结构示意图；
图 6 是本说明书一个实施例提供的一种计算设备的结构框图。

20

具体实施方式

在下面的描述中阐述了很多具体细节以便于充分理解本说明书。但是本说明书能够以很多不同于在此描述的其它方式来实施，本领域技术人员可以在不违背本说明书内涵
25 的情况下做类似推广，因此本说明书不受下面公开的具体实施的限制。

在本说明书一个或多个实施例中使用的术语是仅仅出于描述特定实施例的目的，而非旨在限制本说明书一个或多个实施例。在本说明书一个或多个实施例和所附权利要求书中所使用的单数形式的“一种”、“所述”和“该”也旨在包括多数形式，除非上下文清楚地表示其他含义。还应当理解，本说明书一个或多个实施例中使用的术语“和/
30 或”是指并包含一个或多个相关联的列出项目的任何或所有可能组合。

应当理解，尽管在本说明书一个或多个实施例中可能采用术语第一、第二等来描述各种信息，但这些信息不应限于这些术语。这些术语仅用来将同一类型的信息彼此区分开。例如，在不脱离本说明书一个或多个实施例范围的情况下，第一也可以被称为第二，类似地，第二也可以被称为第一。取决于语境，如在此所使用的词语“如果”可以被解释成为“在……时”或“当……时”或“响应于确定”。

首先，对本说明书一个或多个实施例涉及的名词术语进行解释。

事务：Transaction，由多条 SQL 语句组成的一个执行单元，事务具有原子性、一致性、隔离性、持久性 4 个特性。

MVCC：Multi-Version Concurrency Control，多版本并发控制，一种数据库常用的并发控制机制，每条数据保存多个版本，事务的读请求总是读取对其可见的版本，不会阻塞写请求。

一致性读：Consistent Read，在 MVCC 中，事务可见的数据可以看作数据库在某一时刻的一个“快照”或“视图”，本说明书中所说的一致性特指如下问题：在同一事务内，在备库读到的快照必须和主库读到的快照相同。

时间戳：Timestamp，通常是一个 64 位整数，用于比较判定事件之间的时间先后关系。

DN：Data Node，数据存储节点，通常整个数据库以分片（Sharding）的形式分布在多个 DN 上。

CN：Coordinator Node，事务协调者节点，负责在一个或多个 DN 上协调执行事务。

事务 ID：Transaction ID，简称：trx_id，每个事务的唯一 ID

读取时间戳：Snapshot Timestamp，snapshot_seq，事务的读取时间戳，按照该时间戳判断数据版本的可见性。

提交时间戳：Commit Timestamp，commit_seq，事务的提交。

GTS：Global TimeStamp Service，全局时间戳服务，它负责发放时间戳，本说明书实施例的数据读取装置中的其他节点都向 GTS 请求时间戳。

在本说明书中，提供了一种数据读取方法。本说明书一个或者多个实施例同时涉及一种数据读取装置，一种计算设备，以及一种计算机可读存储介质，在下面的实施例中逐一进行详细说明。

参见图 1，图 1 示出了根据本说明书一个实施例提供的一种数据读取装置的示意图，包括：客户端 102、事务协调模块 104（CN）、时间戳分发模块 106（GTS）、主库 108、

备库 110;

具体实施时, CN 接收客户端 102 发送的读事务请求, 例如查询某些数据的事务请求, 且本实施例中的读事务请求与下述实施例的数据读取事务请求想要表述的内容一致, 其中, 该读事务由一个或多个 SQL 语句组成, 通过该 SQL 语句可以明确读事务请求读取或
5 查询哪些具体的数据;

CN 同时向 GTS 发送为该读事务请求获取对应的读取时间戳的请求, 在接收到 GTS 返回的该读事务的读取时间戳后, 将携带有该读取时间戳的读事务请求发送至与该读取事务请求对应的每个备库 110, 且获取所有备库 110 写入事务的最大提交时间戳, 其中, 该提交时间戳即为主库 108 向对应的备库 110 执行写入事务的最大提交时间戳, 且该提交时间戳根据主库 108 执行的写事务进行递增; 而具体的, 主库 108 向备库 110 执行写
10 事务的时候, 也是通过 CN 进行调度, 且每次写事务执行时, CN 也会向 GTS 发送获取时间戳的请求, 为写事务获取对应的提交时间戳等;

在确定该读事务请求获取对应的读取时间戳小于所有备库 110 写入事务的最大提交时间戳的情况下, 可以确定该读事务请求在主库 108 的数据在该读取时间戳之前已经全部写入备库 110 中, 那么基于该读取时间戳的读事务请求在主库 108 和备库 110 读取的数据就可以保持一致, 且无需引入其他额外的网络请求, 优化网络资源, 且仅通过获取一次读事务的读取时间戳就可以实现基于该读取时间戳的读事务请求从备库 110 中获取到的数据与该读取时间戳的读事务请求从主库 108 中获取的数据保持一致, 提升用户体验。
15

参见图 2, 图 2 示出了根据本说明书一个实施例提供的一种数据读取方法的流程图, 包括以下步骤:

步骤 202: 接收数据读取事务请求。

具体的, 本说明书实施例提供的所述数据读取方法应用于分布式数据库, 且具体应用在 MVCC 多版本并发控制机制中, 而具体的所述数据读取方法的执行对象为上述实施例的 CN。
25

其中, 数据读取事务请求可以理解为待进行数据读取事务的请求, 而数据读取事务则由一条、两条或多条 SQL 语句组成, 例如 SELECT MAX(sal), MIN(age), AVG(sal), SUM(sal) FROM emp 等, 且在实际应用中, 数据读取事务请求一般为客户端发送的。

步骤 204: 基于所述数据读取事务请求获取读取时间戳。

具体的, CN 在接收到客户端发送的数据读取事务请求后, 基于该数据读取事务向 GTS
30

发送为该数据读取事务获取读取时间戳的请求，GTS 为该数据读取事务生成读取时间戳，且将该读取时间戳返回至 CN；而为了便于理解，以下均以读取时间戳为 snapshot_seq 为例，对所述数据读取方法进行详细介绍。

具体实施时，CN 在接收 GTS 返回的该数据读取事务的读取时间戳的同时，会获取第
5 一数据库的写入事务的最大提交时间戳，实际应用中，第一数据库可以理解为备库，即主库的备份数据库，而第一数据库的写入事务的最大提交时间戳，则可以理解为主库向备库写入数据的写入事务最大的提交时间戳。

实际应用中，数据的读取和写入是互不影响的，在进行数据读取的时候不会影响数据的写入，例如在执行数据查询的时候，主库中的数据会同步向备库进行备份，不会因
10 为执行了数据查询事务而终止数据备份，而主库中的数据每向备库写入一次，就相当于执行了一次写事务，CN 在调度主库中的数据写入备库时会在 GTS 产生一个写入事务的提交时间戳。

且由于所述数据读取方法应用于分布式数据库中，因此会存在多个分片的主库和对应的多个分片的备库，CN 在将主库的数据向备库备份时，会将多个分片的主库中的数据
15 向对应的多个分片的备库进行备份，此时就会根据待备份的数据的量的不同，完成该写事务的时间也会不同，因此会产生多个提交时间戳，而本说明书实施例中，获取的第一数据库的写入事务的最大提交时间戳，即是所有的备库的写入事务的最大提交时间戳。

步骤 206：在所述读取时间戳小于第一数据库的写入事务的最大提交时间戳的情况下，将携带有所述读取时间戳的数据读取事务请求发送至所述第一数据库进行数据读取。
20 其中，在读取时间戳小于备库的写入事务的最大提交时间戳的情况下，将携带有读取时间戳的数据读取事务请求发送至第一数据库进行数据读取，并接收第一数据库基于该携带有读取时间戳的数据读取事务请求读取的数据。

具体的，所述读取时间戳小于第一数据库的写入事务的最大提交时间戳包括：

所述读取时间戳小于第二数据库针对所述第一数据库的写入事务的最大提交时间
25 戳，其中，所述第一数据库为所述第二数据库的备份数据库。

其中，第二数据库即可以理解为上述实施例的主库。

本说明书实施例中，所述数据读取方法将读取时间戳与备库的写入事务的最大提交时间戳进行比对，基于该读取时间戳就可以实现在同一个数据读取事务请求下从主库和备库获取的数据一致，提升用户体验。

30 举例说明，若客户端发送的数据读取事务请求的读取时间戳为****12:05:02，而第

一数据库的写入事务的最大提交时间戳为****12:05:03，则可以确定在接收到客户端的该数据读取事务请求时，主库中在****12:05:02 之前的数据已经全部备份到了备库中，此时根据该数据读取事务请求在备库获取到的时间戳为****12:05:02 之前的数据则与在主库获取到的时间戳为****12:05:02 之前的数据保持一致，即主库写入到备库数据的完成时间早于备库执行数据读取事务请求的时间，就可以实现在该数据读取事务请求在主库以及备库读取的数据一致。

本说明书实施例提供的所述数据读取方法包括：接收数据读取事务请求；基于所述数据读取事务请求获取读取时间戳；在所述读取时间戳小于第一数据库的写入事务的最大提交时间戳的情况下，将所述数据读取事务请求发送至所述第一数据库进行数据读取，且接收所述第一数据库基于所述数据读取事务请求读取的数据；所述数据读取方法通过获取一次读取时间戳，基于该读取时间戳就可以保证同一数据读取事务请求下读取的分布式备库的数据与对应的主库的数据一致性，提升了用户在分布式备库进行数据的读取及查询体验。

本说明书另一实施例中，所述基于所述数据读取事务请求获取读取时间戳之后，还包括：

在所述读取时间戳大于所述第一数据库的写入事务的最大提交时间戳的情况下，在等待预设时长后重新获取所述第一数据库的写入事务的最大提交时间戳。

其中，预设时长可以根据实际应用进行设置，在此不做任何限定，例如设置为 1ms、2ms 等。

具体的，为了避免在读取时间戳大于第一数据库的写入事务的最大提交时间戳时，直接结束该数据读取事务请求给用户造成不好的体验，会在读取时间戳大于提交时间戳的情况下，设置一个数据库分布库可接受的等待时间，在等待预设时长后重新获取第一数据库的写入事务的最大提交时间戳，再进行判断，在不影响数据读取事务请求处理的情况下，通过两次的读取时间戳判断保证主库和备库的读取数据的一致性；在第一次通过读取时间戳判断出主库和备库数据不一致的情况下，为客户端的数据读取事务请求争取了第二次的判断机会，避免直接将客户端的数据读取事务请求结束，使得用户在此种情况下需要多次重新提出数据读取事务请求，增加用户的数据读取时间。

具体实施时，在等待预设时长后重新获取所述第一数据库的写入事务的最大提交时间戳后，再次判断读取时间戳是否小于第一数据库的写入事务的最大提交时间戳，若是，则继续执行上述流程，若否，为了保证该数据读取事务请求的正常进行，会通过日志序

列号的方式保证客户端在同一数据读取事务请求下载备库获取的数据与该数据读取事务请求在主句获取的数据一致，具体实现方式如下所述：

所述在等待预设时长后重新获取所述第一数据库的写入事务的最大提交时间戳之后，还包括：

- 5 在所述读取时间戳小于所述第一数据库的写入事务的最大提交时间戳的情况下，获取所述第二数据库的日志序列号；

将携带所述第二数据库的日志序列号的所述数据读取事务请求发送至所述第一数据库，且获取所述第一数据库的日志序列号；

- 10 在确定所述第一数据库的日志序列号与所述第二数据库的日志序列号匹配的情况下，基于所述数据读取事务请求以及所述第一数据库的日志序列号在所述第一数据库读取数据。

其中，日志序列号英文全称为 Log Sequence Number，英文简称为 LSN，是事务日志里面每条记录的唯一编号，主库中数据每执行一次写入事务提交，均会在日志中产生记录，而该记录均会生成对应 LSN，且 LSN 就会同时存储在文件本身及对应的数据表中。

- 15 具体的，在接收数据读取事务请求，且确定该数据读取事务请求的读取时间戳大于备库的写入事务的最大提交时间戳的情况下，获取主库的日志序列号，然后将携带有主库的日志序列号的数据读取事务请求分别发送至与主库对应的备库，然后获取备库的日志序列号，在主库的日志序列号与备库的日志序列号匹配的情况下，基于所述数据读取事务请求以及所述第一数据库的日志序列号在所述第一数据库读取数据。

- 20 实际应用中，主库中的数据会同步备份到备库，那么数据的日志序列号也会同步备份到备库，若备库中存在与主库中的数据相同的日志序列号，则可以说明主库中该日志序列号的数据已经全部备份到了备库，实现主库和备库的数据一致，此时可以基于该数据读取事务请求以及备库的日志序列号在备库读取与主库一致的数据，例如此时主库中存在日志序列号 1-300 的数据，而备库中也存在日志序列号 1-300 的数据，那么该数据
25 读取事务请求读取的主库和备库的数据就会是一致的。

具体实施时，所述获取所述第二数据库的日志序列号包括：

获取所述第二数据库最后一次数据库完整备份的日志序列号；

相应的，所述获取所述第一数据库的日志序列号包括：

获取所述第一数据库最后一次数据库完整备份的日志序列号。

- 30 具体的，获取所述第二数据库最后一次数据库完整备份的日志序列号以及获取所述

第一数据库最后一次数据库完整备份的日志序列号均是为了保证备库中的数据是主库最后一次备份的数据，以保证数据的实时性。

而由于本实施例的所述数据读取方法应用于分布式数据库，因此一个数据读取事务请求可能对应多个主库以及多个备库，因此在接收到客户端发送的数据读取事务请求后，
5 会先对数据读取事务请求进行解析，确定出对应的主库，以便后续的在对应的主库中进行数据处理，具体实现方式如下所述：

所述接收数据读取事务请求之后，还包括：

基于所述数据读取事务请求确定对应的至少一个所述第二数据库。

具体实施时，所述获取所述第二数据库的日志序列号包括：

10 获取每个第二数据库的日志序列号；

相应的，所述将携带所述第二数据库的日志序列号的所述数据读取事务请求发送至所述第一数据库包括：

将携带有所述第二数据库的日志序列号的所述数据读取事务请求发送至与每个第二数据库对应的第一数据库。

15 具体的，在基于所述数据读取事务请求确定对应的至少一个主库后，获取每个主库的日志序列号，然后将每个主库的日志序列号分别与数据读取事务请求进行绑定，然后将绑定了主库的日志序列号的数据读取事务请求分别发送到每个主库对应的备库上，使得备库可以基于该日志序列号处理数据读取事务请求，保证该数据读取事务请求在备库读取的数据与主库的数据保持一致，避免造成数据遗漏。

20 实际应用中，为了保证数据库的运行速度，数据的读取经常会在备库进行，数据的写入会在主库进行，而在接收到客户端的数据读取事务请求后，会向备库中进行数据读取，此时就要保证该数据读取事务请求在主库上读取到的数据和在备库上读取到数据是一致的，在这种情况下，该数据读取事务请求在备库上读取到的数据才可以被正常使用，因此采用对读取时间戳的判断或日志序列号的方式来保证同一个数据读取事务请求在备
25 库读取到的数据与该数据读取事务请求在主库读取的数据是一致的，增强用户体验。

本说明书实施例提供的所述数据读取方法巧妙设计和利用 GTS 时间戳的单调性，在全局分片上建立因果关系，只需要获取一次时间戳就可以满足所有分片的主备一致性判断，体现出好的网络优化，节省网络资源。

以下对采用所述数据读取方法中若数据读取事务的读取时间戳大于第一数据的写入
30 事务的最大提交时间戳的情况下，在等待预设时长后重试上述步骤可以满足数据读取，

进行正确性验证:

假设存在一个写事务 T1, 即主库之前写入过的事务, 通过同步链路同步到备库上, 也可以看做待读取的数据、一个读事务 S; 那么实际应用中, T1 应该对 S 可见, 即在备库上 T1 要么对 S 可见, 要么 S 阻塞等待, 然后根据时间戳判断可见。

- 5 若 $T1.commit_seq < S.snapshot_seq$, $S.snapshot_seq < max_commit_seq$, 即存在一个提交事件 $T2.commit_seq > S.snapshot_seq$; 具体的 T1 的提交时间戳小于 S 的读取时间戳, S 的读取时间戳小于备库的写入事务的最大提交时间戳, 且该 max_commit_seq 由写事务 T2 获得;

- 10 那么通过 $T1.commit_seq < S.snapshot_seq$, 可知 $T1.commit_seq < T2.commit_seq$, 所以 $T1.GetCommitTS \rightarrow T2.GetCommitTS$ ($\rightarrow$ 表示 happens-before); 具体的, 由于 T1 的提交时间戳小于 S 的读取时间戳, S 的读取时间戳又小于 T2 的提交时间戳, 那么可以认为 T1 的提交时间戳小于 T2 的提交时间戳, 即 T1 获取提交时间戳在 T2 获取提交时间戳之前;

而事务提交流程是 Prepare $\rightarrow$ GetCommitTS $\rightarrow$ Commit, 所以有:

- 15 $T1.Prepare \rightarrow T1.GetCommitTS \rightarrow T1.Commit$
 $T2.Prepare \rightarrow T2.GetCommitTS \rightarrow T2.Commit$;

具体的, 事务提交流程是准备、获取提交时间戳、提交, 因此本次事务提交流程为:

T1 准备、获取 T1 的提交时间戳、T1 提交;

T2 准备、获取 T2 的提交时间戳、T2 提交;

- 20 而根据上述可知, T1 准备、获取 T1 的提交时间戳在 T2 准备、获取 T2 的提交时间戳之前, 但是分布式系统会存在时延, 所以 T1 的提交时间戳以及 T2 的提交时间戳到达备库的顺序可能会存在乱序, 因此 T1 提交和 T2 提交这两个事件的先后关系仍是不确定的;

- 25 即备库 Apply 日志必然会保证上述 T1 和 T2 的事务提交的先后关系保持不变, 因此 S 创建时 $T1.Prepare$ 一定已经完成了, 那么 $T1.Commit$ 的时序有两种情况:

如果 $T1.Commit$ 先于 $T2.Commit$, 那么 T1 直接对 S 可见,

如果 $T1.Commit$ 后于 $T2.Commit$, 那么 S 会阻塞等待 $T1.Commit$ 到达备库, 然后根据时间戳判断出可见。

- 30 具体的, 即 T1 的提交早于 T2 的情况下, 在 T2 为最后一次写入事务的情况下, T1 对 S 是可见的, 而在 T1 的提交晚于 T2 的情况下, 但 T1 的提交时间戳获取在 T2 的提交

时间戳获取之前，且 T2.commit_seq 为备库的写入事务的最大提交时间戳，但是 T1 的提交晚于 T2 提交，此时则 S 会阻塞等待 T1.Commit 到达备库，再根据备库的写入事务的最大提交时间戳确定 T1 对 S 可见，以保证读取备库数据的完整性。

本说明书实施例提供的所述数据读取方法，在备库节点维护目前最大的提交时间戳 max_commit_seq，如果读事务 S 的读取时间戳满足 snapshot_seq < max_commit_seq，则可以认为读是“安全”的，该备库中的写事务对读事务 S 可见，如果不满足本前提，则可以阻塞等待一小段时间（例如 1ms）之后重试，在备库的写入事务的最大提交时间戳大于读事务 S 的读取时间戳的情况下，仍可以确定备库中的写事务对读事务 S 可见，通过阻塞等待的方式快速安全的满足读事务读取数据的请求。

10 下述结合附图 3，以本说明书提供的所述数据读取方法在分布式数据库的应用为例，对所述数据读取方法进行进一步说明。其中，图 3 示出了本说明书一个实施例提供的第一种数据读取方法的处理过程流程图，具体包括以下步骤：

具体实施时，本说明书实施例提供的所述数据读取方法中包括用户、CN、GTS 以及分布式备库 DN1\DN2\DN3。

15 步骤 302：用户向 CN 发送数据读取事务请求。

步骤 304：CN 接收到数据读取事务请求后向请求 GTS 为数据读取事务生成读取时间戳以及获取分布式备库 DN1\DN2\DN3 的写入事务的最大提交时间戳。

步骤 306：GTS 为数据读取事务生成读取时间戳以及获取分布式备库 DN1\DN2\DN3 的写入事务的最大提交时间戳后，将该读取时间戳以及提交时间戳返回至 CN。

20 步骤 308：CN 在确定读取时间戳小于提交时间戳的情况下，将携带有该读取时间戳的数据读取事务请求分别发送至对应的分布式备库 DN1\DN2\DN3。

步骤 310：分布式备库 DN1\DN2\DN3 基于数据读取事务请求读取的数据，且将读取的数据返回至 CN。

步骤 312：CN 将接收的数据返回给用户。

25 本说明书实施例提供的所述数据读取方法，通过获取一次时间戳，基于该时间戳可以保证读取的分布式备库的数据与对应的主库的数据一致，极大的提升了用户在分布式备库进行数据的读取及查询体验。

下述结合附图 4，以本说明书提供的所述数据读取方法在分布式数据库的应用为例，对所述数据读取方法进行进一步说明。其中，图 4 示出了本说明书一个实施例提供的第
30 二种数据读取方法的处理过程流程图，具体包括以下步骤：

具体实施时，本说明书实施例提供的数据库读取方法中包括用户、CN、GTS 以及分布式备库 DN1\DN2\DN3、分布式主库 MDB1\MDB2\MDB3。

步骤 402：用户向 CN 发送数据库读取事务请求。

步骤 404：CN 接收到数据库读取事务请求后向请求 GTS 为数据库读取事务生成读取时间戳以及获取分布式备库 DN1\DN2\DN3 的写入事务的最大提交时间戳。

步骤 406：GTS 为数据库读取事务生成读取时间戳以及获取分布式备库 DN1\DN2\DN3 的写入事务的最大提交时间戳后，将该读取时间戳以及提交时间戳返回至 CN。

步骤 408：CN 在确定读取时间戳大于分布式备库 DN1\DN2\DN3 的写入事务的最大提交时间戳的情况下，在等待预设时长后重新获取所述第一数据库的写入事务的最大提交时间戳，在读取时间戳仍大于分布式备库 DN1\DN2\DN3 的写入事务的最大提交时间戳的情况下，分别向分布式主库 MDB1\MDB2\MDB3 发送获取日志序列号的请求。

步骤 410：分布式主库 MDB1\MDB2\MDB3 接收到日志序列号的获取请求后分别将其对应的日志序列号返回至 CN。

步骤 412：CN 将携带有对应的日志序列号的数据库读取事务请求分别发送至对应的分布式备库 DN1\DN2\DN3。

步骤 414：分布式备库 DN1\DN2\DN3 接收到携带有日志序列号的数据库读取事务请求后，等待本地日志 Apply 的日志序列号到达当前接收的数据库读取事务请求中携带的日志序列号时，基于数据库读取事务请求进行数据库读取且将读取到的数据返回至 CN。

步骤 416：CN 将接收到的数据返回给用户。

本说明书实施例提供的所述数据库读取方法，在备库 DN 节点维护目前最大的提交时间戳（max_commit_seq），在备库 DN 节点上，如果读取时间戳满足 snapshot_seq>max_commit_seq，则可以认为读是“安全”的，可以直接进行数据库读取，如果不满足本前提，可以等待一小段时间（例如 1ms）之后重试，即重新获取备库 DN 节点维护目前写入事务的最大的提交时间戳（max_commit_seq），若仍不满足，则先到主库取当前 LSN，然后等待备库 LSN 到达当前 LSN 时在备库进行数据库读取，以保证读取的备库数据与主库一致，通过三种方式实现读取备库数据时保证基于同一个数据库读取事务在备库读取到的数据和在主库读取到的数据保持一致，提升用户的数据库读取体验，且备库的读一般都可以在一次 RT（round-trip，一次读请求）内完成，提高了数据的查询性能。

与上述方法实施例相对应，本说明书还提供了数据库读取装置实施例，图 5 示出了本

说明书一个实施例提供的一种数据读取装置的结构示意图。如图 5 所示，该装置包括：

请求接收模块 502，被配置为接收数据读取事务请求；

时间戳获取模块 504，被配置为基于所述数据读取事务请求获取读取时间戳；

- 5 数据读取模块 506，被配置为在所述读取时间戳小于第一数据库的写入事务的最大提交时间戳的情况下，将携带有所述读取时间戳的数据读取事务请求发送至所述第一数据库进行数据读取。

可选的，所述数据读取模块 506，进一步被配置为：

所述读取时间戳小于第二数据库针对所述第一数据库的写入事务的最大提交时间戳，其中，所述第一数据库为所述第二数据库的备份数据库。

- 10 可选的，所述装置，还包括：

重试模块，被配置为在所述读取时间戳大于所述第一数据库的写入事务的最大提交时间戳的情况下，在等待预设时长后重新获取所述第一数据库的写入事务的最大提交时间戳。

可选的，所述装置，还包括：

- 15 第一序列号获取模块，被配置为在所述读取时间戳小于所述第一数据库的写入事务的最大提交时间戳的情况下，获取所述第二数据库的日志序列号；

第二序列号获取模块，被配置为将携带所述第二数据库的日志序列号的所述数据读取事务请求发送至所述第一数据库，且获取所述第一数据库的日志序列号；

- 20 序列号匹配模块，被配置为在确定所述第一数据库的日志序列号与所述第二数据库的日志序列号匹配的情况下，基于所述数据读取事务请求以及所述第一数据库的日志序列号在所述第一数据库读取数据。

可选的，所述第一序列号获取模块，进一步被配置为：

获取所述第二数据库最后一次数据库完整备份的日志序列号；

相应的，所述第二序列号获取模块，进一步被配置为：

- 25 获取所述第一数据库最后一次数据库完整备份的日志序列号。

可选的，所述装置，还包括：

数据库确定模块，被配置为基于所述数据读取事务请求确定对应的至少一个所述第二数据库。

可选的，所述第一序列号获取模块，进一步被配置为：

- 30 获取每个第二数据库的日志序列号；

相应的，第二序列号获取模块，进一步被配置为：

将携带有所述第二数据库的日志序列号的所述数据读取事务请求发送至与每个第二数据库对应的第一数据库。

可选的，所述数据读取方法应用于分布式数据库。

- 5 本说明书一个实施例提供了一种数据读取装置包括接收数据读取事务请求；基于所述数据读取事务请求获取读取时间戳；在所述读取时间戳小于第一数据库的写入事务的最大提交时间戳的情况下，将所述数据读取事务请求发送至所述第一数据库进行数据读取，且接收所述第一数据库基于所述数据读取事务请求读取的数据；所述数据读取装置通过获取一次时间戳，基于该时间戳就可以保证同一数据读取事务请求下读取的分布式
- 10 备库的数据与对应的主库的数据一致性，极大的提升了用户在分布式备库进行数据的读取及查询体验。

上述为本实施例的一种数据读取装置的示意性方案。需要说明的是，该数据读取装置的技术方案与上述的数据读取方法的技术方案属于同一构思，数据读取装置的技术方案未详细描述的细节内容，均可以参见上述数据读取方法的技术方案的描述。

- 15 图 6 示出了根据本说明书一个实施例提供的一种计算设备 600 的结构框图。该计算设备 600 的部件包括但不限于存储器 610 和处理器 620。处理器 620 与存储器 610 通过总线 630 相连接，数据库 650 用于保存数据。

- 计算设备 600 还包括接入设备 640，接入设备 640 使得计算设备 600 能够经由一个或多个网络 660 通信。这些网络的示例包括公用交换电话网（PSTN）、局域网（LAN）、
- 20 广域网（WAN）、个域网（PAN）或诸如因特网的通信网络的组合。接入设备 640 可以包括有线或无线的任何类型的网络接口（例如，网络接口卡（NIC））中的一个或多个，诸如 IEEE802.11 无线局域网（WLAN）无线接口、全球微波互联接入（Wi-MAX）接口、以太网接口、通用串行总线（USB）接口、蜂窝网络接口、蓝牙接口、近场通信（NFC）接口，等等。

- 25 在本说明书的一个实施例中，计算设备 600 的上述部件以及图 6 中未示出的其他部件也可以彼此相连接，例如通过总线。应当理解，图 6 所示的计算设备结构框图仅仅是出于示例的目的，而不是对本说明书范围的限制。本领域技术人员可以根据需要，增添或替换其他部件。

- 计算设备 600 可以是任何类型的静止或移动计算设备，包括移动计算机或移动计算
- 30 设备（例如，平板计算机、个人数字助理、膝上型计算机、笔记本电脑、上网本等）、

移动电话（例如，智能手机）、可佩戴的计算设备（例如，智能手表、智能眼镜等）或其他类型的移动设备，或者诸如台式计算机或 PC 的静止计算设备。计算设备 600 还可以是移动式或静止式的服务器。

其中，处理器 620 用于执行如下计算机可执行指令，其中，所述处理器执行所述计算机可执行指令时实现所述数据读取方法的步骤。

上述为本实施例的一种计算设备的示意性方案。需要说明的是，该计算设备的技术方案与上述的数据读取方法的技术方案属于同一构思，计算设备的技术方案未详细描述的细节内容，均可以参见上述数据读取方法的技术方案的描述。

本说明书一实施例还提供一种计算机可读存储介质，其存储有计算机指令，该指令被处理器执行时实现所述数据读取方法的步骤。

上述为本实施例的一种计算机可读存储介质的示意性方案。需要说明的是，该存储介质的技术方案与上述的数据读取方法的技术方案属于同一构思，存储介质的技术方案未详细描述的细节内容，均可以参见上述数据读取方法的技术方案的描述。

上述对本说明书特定实施例进行了描述。其它实施例在所附权利要求书的范围内。在一些情况下，在权利要求书中记载的动作或步骤可以按照不同于实施例中的顺序来执行并且仍然可以实现期望的结果。另外，在附图中描绘的过程不一定要示出的特定顺序或者连续顺序才能实现期望的结果。在某些实施方式中，多任务处理和并行处理也是可以的或者可能是有利的。

所述计算机指令包括计算机程序代码，所述计算机程序代码可以为源代码形式、对象代码形式、可执行文件或某些中间形式等。所述计算机可读介质可以包括：能够携带所述计算机程序代码的任何实体或装置、记录介质、U 盘、移动硬盘、磁碟、光盘、计算机存储器、只读存储器（ROM, Read-Only Memory）、随机存取存储器（RAM, Random Access Memory）、电载波信号、电信号以及软件分发介质等。需要说明的是，所述计算机可读介质包含的内容可以根据司法管辖区内立法和专利实践的要求进行适当的增减，例如在某些司法管辖区，根据立法和专利实践，计算机可读介质不包括电载波信号和电信信号。

需要说明的是，对于前述的各方法实施例，为了简便描述，故将其都表述为一系列的动作组合，但是本领域技术人员应该知悉，本说明书实施例并不受所描述的动作顺序的限制，因为依据本说明书实施例，某些步骤可以采用其它顺序或者同时进行。其次，本领域技术人员也应该知悉，说明书中所描述的实施例均属于优选实施例，所涉及的动

作和模块并不一定都是本说明书实施例所必须的。

在上述实施例中，对各个实施例的描述都各有侧重，某个实施例中未详述的部分，可以参见其它实施例的相关描述。

5 以上公开的本说明书优选实施例只是用于帮助阐述本说明书。可选实施例并没有详尽叙述所有的细节，也不限制该发明仅为所述的具体实施方式。显然，根据本说明书实施例的内容，可作很多的修改和变化。本说明书选取并具体描述这些实施例，是为了更好地解释本说明书实施例的原理和实际应用，从而使所属技术领域技术人员能很好地理解和利用本说明书。本说明书仅受权利要求书及其全部范围和等效物的限制。