

一种应用于分布式数据库的时间戳分配方法及装置

技术领域

[01] 本说明书一个或多个实施例涉及数据库技术领域，尤其涉及一种应用于分布式数据库的时间戳分配方法及装置。

背景技术

[02] 分布式数据库作为分布式系统中的重要应用部分，不仅继承了传统单机数据库的核心特性，同时还拥有分布式系统的处理能力，具有低硬件成本、高可扩展性、高可用性等特性。分布式数据库由多个节点组成，各个节点需要通过交换消息来相互通信。由于每个节点上的数据库实例拥有独立的时钟，而每个节点上的时钟不能反映全局的顺序，因此会存在时钟偏移、无法保证分布式数据库的一致性的情况。

[03] 为了避免发生时钟偏移，可以通过时钟服务来提供全局一致的版本号。时钟服务即利用时间戳（timestamp，ts）为数据库中的事务建立一个先后关系，时间戳越大表示事务的版本号越大，该事务的顺序越靠后。相关技术中通过单一的授时节点统一给数据库中的事务提供时钟服务，单一授时节点可以保证按照递增的方式给各节点的事务分配时间戳，从而确保数据库中事务的一致性。

[04] 对于此类使用单点授时时钟服务的分布式数据库来说，所有节点针对每个事务都需要向该授时节点发送获取时间戳的请求，容易导致该授时节点负载过高，当达到授时节点的处理上限时，会大大增加时间戳获取请求的响应时间。再者，对各个事务而言，每次提交时都需要向授时节点发送请求，并等待授时节点返回分配的时间戳，会大大增加各个事务处理的时间，降低数据库性能。

发明内容

[05] 本说明书提供一种应用于分布式数据库的时间戳分配方法及装置，以解决相关技术中的不足。

[06] 根据本说明书一个或多个实施例的第一方面，提供一种应用于分布式数据库的时间戳分配方法，所述分布式数据库的节点包括授时节点和普通节点，所述方法应用于任一节点，该方法包括：响应于接收的事务提交请求，提交相应的待授时事务并记录所述待

授时事务的开始提交时刻；查询所述任一节点针对其他事务向所述授时节点发送授时请求的授时请求发送时刻；在任一其他事务的授时请求发送时刻不早于所述待授时事务的开始提交时刻的情况下，将所述授时节点针对该任一其他事务反馈的时间戳分配至所述待授时事务。

[07]根据本说明书一个或多个实施例的第二方面，提供一种应用于分布式数据库的时间戳分配装置，所述分布式数据库的节点包括授时节点和普通节点，所述装置应用于任一节点，该装置包括：提交模块，用于响应于接收的事务提交请求，提交相应的待授时事务并记录所述待授时事务的开始提交时刻；查询模块，用于查询所述任一节点针对其他事务向所述授时节点发送授时请求的授时请求发送时刻；分配模块，用于在任一其他事务的授时请求发送时刻不早于所述待授时事务的开始提交时刻的情况下，将所述授时节点针对该任一其他事务反馈的时间戳分配至所述待授时事务。

[08]根据本说明书一个或多个实施例的第三方面，提供一种电子设备，包括：处理器；用于存储处理器可执行指令的存储器；其中，所述处理器通过运行所述可执行指令以实现如上述第一方面的实施例中所述的方法。

[09]根据本说明书一个或多个实施例的第四方面，提供一种计算机可读存储介质，其上存储有计算机指令，该指令被处理器执行时实现如上述第一方面的实施例中所述方法的步骤。

[10]由以上技术方案可见，本说明书一个或多个实施例中，通过在分布式数据库的各个节点增加时间戳缓存来保存曾经从授时节点中获取到的时间戳，使得在保证数据库外部一致性的基础上，多个事务可以复用同一个时间戳，从而：一方面，有效降低了各节点向授时节点发送获取时间戳请求的频率，相应的，减少了授时节点处理请求所产生的负载，同时也避免了因授时节点过载导致授时节点返回授时响应过久的问题；另一方面，由于避免了等待授时节点分配时间戳，所以缩短了等待时长，从而大大提高了各个事务的处理效率，保证了数据库的性能。

[11]应当理解的是，以上的一般描述和后文的细节描述仅是示例性和解释性的，并不能限制本说明书。

附图说明

[12]此处的附图被并入说明书中并构成本说明书的一部分，示出了符合本说明书的实施

例，并与说明书一起用于解释本说明书的原理。

[13]图 1 是一示例性实施例提供的一种分布式数据库的架构图。

[14]图 2 是一示例性实施例提供的一种应用于分布式数据库的时间戳分配方法的流程图。

[15]图 3a 是一示例性实施例提供的一种保证分布式数据库外部一致性的示意图。

[16]图 3b 是另一示例性实施例提供的一种保证分布式数据库外部一致性的示意图。

[17]图 4 是一示例性实施例提供的一种电子设备的结构示意图。

[18]图 5 是一示例性实施例提供的一种应用于分布式数据库的时间戳分配装置的框图。

具体实施方式

[19]这里将详细地对示例性实施例进行说明，其示例表示在附图中。下面的描述涉及附图时，除非另有表示，不同附图中的相同数字表示相同或相似的要素。以下示例性实施例中所描述的实施方式并不代表与本发明说明书相一致的所有实施方式。相反，它们仅是与如所附权利要求书中所详述的、本发明说明书的一些方面相一致的装置和方法的例子。

[20]需要说明的是：在其他实施例中并不一定按照本发明说明书示出和描述的顺序来执行相应方法的步骤。在一些其他实施例中，其方法所包括的步骤可以比本发明说明书所描述的更多或更少。此外，本发明说明书中所描述的单个步骤，在其他实施例中可能被分解为多个步骤进行描述；而本发明说明书中所描述的多个步骤，在其他实施例中也可能被合并为单个步骤进行描述。

[21]分布式数据库因其高可扩展性、高可靠性而被广泛应用于数据的存储及管理。图 1 是一示例性实施例提供的一种分布式数据库的架构图。如图 1 所示，分布式数据库由多个节点组成，包括授时节点和普通节点。所有普通节点均向授时节点发送授时请求，以获取授时节点分配的时间戳。相关技术中通过此架构下单一授时节点统一给数据库中的事务提供时钟服务的方式来保证数据库中事务的一致性。但这种单点授时的方式，使得各普通节点需要针对本节点上的每个事务逐一向授时节点发送授时请求，容易导致授时节点处理授时请求的负载过高，当达到授时节点的处理上限时，会大大增加授时请求的响应时间。再者，对各个事务而言，每次提交时都需要向授时节点发送授时请求，并等待授时节点返回分配的时间戳，会大大增加各个事务执行的时间，降低数据库性能。

[22]本发明说明书提供一种应用于分布式数据库的时间戳分配方法，以解决相关技术中的不

足。

[23]图2是一示例性实施例提供的一种应用于分布式数据库的时间戳分配方法的流程图。该方法可以应用于图1中分布式数据库的任一节点。除普通节点外，当授时节点不仅具备授时功能、还能实现普通节点对事务的处理功能时，该方法同样适用于授时节点。如图2所示，该方法可以包括以下步骤S201至S203。

[24]S201，响应于接收的事务提交请求，提交相应的待授时事务并记录所述待授时事务的开始提交时刻。

[25]数据库的事务可以理解为数据库执行过程中的一个逻辑单位，由一个有限的数据库操作序列构成。通常一个事务会包含多个读写操作。事务具有四个基本特性：原子性（Atomicity）、一致性（Consistency）、隔离性（Isolation）、持久性（Durability）。事务的持久性决定了事务只有提交才会有效，提交之后，事务对数据库的相关修改就能永久保存在数据库中，被其他用户读取，如果不提交，事务内容会存在缓存中，随时可以回滚。数据库事务有三种提交方式：手动显式提交、隐式提交以及自动提交，本领域技术人员可以根据实际需求自行设置，本说明书对此不做限定。

[26]一般的，事务执行的整个过程可以分为事务开始提交、事务获取时间戳、事务完成提交三个阶段。其中，时间戳的获取过程可以分为普通节点针对提交的事务向授时节点发送授时请求、授时节点接收授时请求并向发送此授时请求的普通节点返回授时响应、普通节点接收授时响应并获取反馈至该事务的时间戳。但这种获取时间戳的过程比较耗时，并且每个事务都需要单独向授时节点发送授时请求的方式容易导致授时节点处理请求的负载过高。基于此，本说明书提出另一种获取时间戳的方式，即多个事务复用同一个时间戳。具体的，当分布式数据库中的任一节点接收到事务提交请求时，提交相应的待授时事务，并记录该待授时事务的开始提交时刻，以便后续过程使用。本说明书中可以将待授时事务的开始提交时刻记作 Absolute Time（Start To Commit）。

[27]S202，查询所述任一节点针对其他事务向所述授时节点发送授时请求的授时请求发送时刻。

[28]通常，在待授时事务开始提交之后，接收该事务提交请求的节点需要向授时节点发送授时请求，以使得授时节点对该待授时事务分配时间戳。因此，会存在一个向授时节点发送授时请求的授时请求发送时刻。通常情况下，事务的开始提交时刻与授时请求发送时刻不是同一时刻，两者存在一定的时间差。本说明书中普通节点与授时节点之间的

信息交互依赖于 RPC (Remote Procedure Call) 协议, 所以可以把授时请求发送时刻记作 Absolute Time (Send RPC Request)。当然, 普通节点与授时节点也可以基于其他协议进行信息交互, 本说明书对此不做限定。

[29]以普通节点 N1 为例, 在一实施例中, 普通节点 N1 接收到事务 T0 的提交请求, 响应于该请求提交事务 T0, 并且记录下事务 T0 的开始提交时刻, 记作 Absolute Time (T0. Start To Commit)。此时事务 T0 为待授时事务, 需要对其分配时间戳。当节点 N1 记录下待授时事务 T0 的开始提交时刻后, 节点 N1 可以查询本节点针对其他事务向授时节点发送授时请求的授时请求发送时刻。这里的其他事务可以理解为节点 N1 上除当前待授时事务 T0 外的所有已经向授时节点发送授时请求的事务, 包括已授时事务和未授时事务。其中, 已授时事务指的是已向授时节点发送授时请求、且在待授时事务 T0 开始提交时已收到授时节点返回授时响应的事务, 而未授时事务指的是已向授时节点发送授时请求、但在待授时事务 T0 开始提交时未收到授时节点返回授时响应的事务。

[30]S203, 在任一其他事务的授时请求发送时刻不早于所述待授时事务的开始提交时刻的情况下, 将所述授时节点针对该任一其他事务反馈的时间戳分配至所述待授时事务。

[31]对于同一个事务而言, 其整个执行过程在时间上的顺序为事务开始提交、向授时节点发送授时请求、授时节点接收授时请求、事务完成提交。也就是说, 事务完成提交的时刻晚于授时节点接收授时请求的时刻, 授时节点接收授时请求的时刻晚于向授时节点发送授时请求的时刻, 向授时节点发送授时请求的时刻晚于事务开始提交的时刻。而数据库的外部一致性指的是事务在数据库内部的执行序列不能违背外部观察到的顺序。如果数据库外部能够观察到两个事务存在先后顺序, 那么一定是前一个事务完成提交后, 后一个事务才开始提交。否则两个事务就是并发的, 数据库内部可以以任意顺序执行这两个事务。

[32]举个例子, 数据库外部观察到第一个事务在一个节点写入数据 A, 完成后, 开始启动第二个事务在另一个节点写入数据 B。在这个过程中另有第三个并发运行的事务尝试读 A 和 B, 如果读到了数据 B 却没有读到数据 A, 那意味着在数据库内部第二个事务先于第一个事务执行了, 这与数据库外部观察到的顺序相反, 即没有满足外部一致性。因此, 为了保证数据库的外部一致性, 需要给每个事务分配时间戳。外部一致性的目标是当事务 Ta 完成提交后, 事务 Tb 才开始启动提交时, 事务 Tb 分配到的时间戳必须大于事务 Ta 分配到的时间戳, 这才能保证数据库内部按照时间戳的大小先执行事务 Ta 再执行事务 Tb。

[33]在一实施例中,事务 T1 完成提交后,事务 T3 开始提交,为了保证外部一致性,需要使得事务 T3 的时间戳大于事务 T1 的时间戳。如果在事务 T3 开始提交时,本节点上存在某一事务的授时请求发送时刻不早于事务 T3 的开始提交时刻,那么当该事务收到授时节点返回的时间戳时,事务 T3 可以复用该事务的时间戳。下面结合图 3 进行详细说明:如图 3a 所示,事务 T2 的授时请求发送时刻 306 晚于事务 T3 的开始提交时刻 309,那么可以推出授时节点接收事务 T2 授时请求的时刻 307 晚于事务 T3 的开始提交时刻 309。又因为事务 T3 是在事务 T1 完成提交后才开始启动提交,即事务 T3 的开始提交时刻 309 晚于事务 T1 的完成提交时刻 304,进一步可以推出授时节点接收事务 T2 授时请求的时刻 307 晚于事务 T1 的完成提交时刻 304,那么,事务 T2 的时间戳一定大于事务 T1 的时间戳。此时,事务 T3 复用事务 T2 的时间戳,即可实现事务 T3 的时间戳大于事务 T1 的时间戳,从而保证了外部一致性。

[34]当然,如果事务 T2 的授时请求发送时刻 306 与事务 T3 的开始提交时刻 309 为同一时刻,同样可以得到授时节点接收事务 T2 授时请求的时刻 307 晚于事务 T3 的开始提交时刻 309,进一步可以推出授时节点接收事务 T2 授时请求的时刻 307 晚于事务 T1 的完成提交时刻 304,即事务 T2 的时间戳大于事务 T1 的时间戳。此时,事务 T3 复用事务 T2 的时间戳,同样可以保证外部一致性。也就是说,在事务 T2 的授时请求发送时刻 306 不早于事务 T3 的开始提交时刻 309 的情况下,事务 T3 复用事务 T2 的时间戳,可以保证事务 T3 与事务 T1 之间的外部一致性。

[35]上述实施例中,事务 T2 的开始提交时刻 305 可以早于事务 T1 的完成提交时刻 304,也可以晚于事务 T1 的完成提交时刻 304,即事务 T2 的开始提交时刻 305 与本说明书中多个事务复用同一个时间戳无关。

[36]此外,上述实施例中,事务 T1 的时间戳可以从授时节点获取,也可以复用其他事务的时间戳,只要该其他事务的授权请求发送时刻不早于事务 T1 的开始提交时刻即可。参考图 3b,事务 T0 的授权请求发送时刻 311 晚于事务 T1 的开始提交时刻 301,根据前述推论,事务 T1 可以复用事务 T0 的时间戳,此时事务 T1 的完成提交时刻 304 一定晚于授时节点接收事务 T0 授时请求的时刻 312。又根据图 3a 中的推论:授时节点接收事务 T2 授时请求的时刻 307 晚于事务 T1 的完成提交时刻 304,所以授时节点接收事务 T2 授时请求的时刻 307 一定晚于授时节点接收事务 T0 授时请求的时刻 312,即事务 T2 的时间戳大于事务 T0 的时间戳。那么,当事务 T3 复用事务 T2 的时间戳、且事务 T1 复用事务 T0 的时间戳时,事务 T3 的时间戳大于事务 T1 的时间戳,由此,实现了事务

T1 和事务 T3 的外部一致性。

[37] 综上，当存在某一事务 Tn 的授时请求发送时刻不早于待授时事务 Tx 的开始提交时刻，即 $\text{Absolute Time (Tn. Send RPC Request)} \geq \text{Absolute Time (Tx. Start To Commit)}$ 时，Tx 使用 Tn 获取的时间戳即可满足外部一致性。

[38] 在一实施例中，节点 N1 可以按照收到授时节点返回时间戳的先后顺序，优先查询本节点针对最新已授时事务向授时节点发送授时请求的授时请求发送时刻。由于最新已授时事务的授时请求发送时刻存储在时间戳缓存中，所以，可以将最新已授时事务的授时请求发送时刻记作 $\text{Absolute Time (ts cache. Last Send RPC Request)}$ 。当最新已授时事务的授时请求发送时刻不早于待授时事务 T0 的开始提交时刻，即 $\text{Absolute Time (ts cache. Last Send RPC Request)} \geq \text{Absolute Time (T0. Start To Commit)}$ 时，将该最新已授时事务接收到的时间戳分配至待授时事务 T0，即待授时事务 T0 与该最新已授时事务复用同一个时间戳，这两个事务的版本号一致。在这种情况下，节点 N1 不需要针对待授时事务 T0 向授时节点发送授时请求，一方面，降低了节点 N1 向授时节点发送授时请求的频率，同时，也相应减少了授时节点的负载压力。另一方面，由于最新已授时事务接收到的时间戳已经存储在缓存中，因此待授时事务 T0 可以直接复用缓存中的时间戳，大大减少了等待授时的时间，从而提高了待授时事务 T0 的处理效率。

[39] 在另一实施例中，当最新已授时事务的授时请求发送时刻不满足预设条件时，节点 N1 可以查询本节点上除最新已授时事务外的其他未授时事务向授时节点发送授时请求的授时请求发送时刻。同样的，将未授时事务的授时请求发送时刻记作 $\text{Absolute Time (ts cache. Send RPC Request)}$ 。当查询到存在任一未授时事务的授时请求发送时刻不早于待授时事务 T0 的开始提交时刻时。即 $\text{Absolute Time (ts cache. Send RPC Request)} \geq \text{Absolute Time (T0. Start To Commit)}$ 时，表示该任一未授时事务对应的授时响应中的时间戳可以分配至待授时事务 T0，使得待授时事务 T0 与该任一未授时事务复用同一个时间戳，这两个事务的版本号一致。比如，查询到存在未授时事务 T4 的授时请求发送时刻满足预设条件，此时，只需将待授时事务 T0 添加至节点 N1 缓存的待授时队列中，等待授时节点返回针对未授时事务 T4 的授时响应。当未授时事务 T4 接收到授时节点返回的授时响应时，将此授时响应中的时间戳分配至待授时事务 T0，使得待授时事务 T0 和事务 T4 复用同一个时间戳。在这种情况下，节点 N1 同样不需要针对待授时事务 T0 向授时节点发送授时请求，降低了节点 N1 向授时节点发送授时请求的频率，同时，也在一定程度上减少了授时节点的负载压力。本实施例中，将待授时事务 T0 添加至待授

时队列以等待未授时事务 T4 的授时响应返回可以作为一个异步任务，从而避免影响其他事务的提交过程。

[40]在一实施例中，在节点 N1 上所有除待授时事务 T0 以外的其他事务的授时请求发送时刻都不满足预设条件，即所有其他事务的授时请求发送时刻都早于待授时事务 T0 的开始提交时刻的情况下，节点 N1 需要针对待授时事务 T0 向授时节点发送授时请求。授时节点响应于接收到的授时请求分配时间戳，将此时间戳赋值在授时响应中，并且授时响应中包含了授时请求的相关信息，以使得节点 N1 在接收到授时响应时，可以明确该授时响应对应哪个授时请求。因为授时节点的时间戳不会回退，即授时节点按照递增的方式给各节点的事务分配时间戳，因此可以保证最新授时事务的时间戳一定是最大的，进而保证了数据库的外部一致性。

[41]在一实施例中，分布式数据库中的各个节点还负责接收授时节点返回的任一授时响应，并将接收到的授时响应对应的授时请求发送时刻与时间戳缓存中最新已授时事务的授时请求发送时刻进行比较。如果授时响应对应的授时请求发送时刻晚于时间戳缓存中最新已授时事务的授时请求发送时刻，那么将缓存中最新已授时事务的授时请求发送时刻更新为授时响应对应的授时请求发送时刻。同时，将时间戳缓存中存储的时间戳更新为该授时响应反馈的时间戳。如果授时响应对应的授时请求发送时刻不晚于时间戳缓存中最新已授时事务的授时请求发送时刻，就丢掉该授时响应。这一更新操作可以保证缓存中最新已授时事务的授时请求发送时刻、即 Absolute Time (ts cache. Last Send RPC Request) 一直是已授时事务的授时请求发送时刻集合中的最大值，进而使得满足事物开始提交时刻不晚于最新已授时事务的授时请求发送时刻这一条件的任一待授时事务都可以使用最新已授时事务的时间戳。

[42]在一实施例中，每次更新完时间戳缓存中存储的时间戳后，遍历该节点上待授时队列的所有待授时事务。因为更新后，缓存中最新已授时事务的授时请求发送时刻是当前缓存中已授时事务的授时请求发送时刻集合中的最大值，所以可以将更新后的时间戳分配至所有事物开始提交时刻不晚于当前最大授时请求发送时刻的待授时事务。

[43]在一实施例中，可以在授时节点返回的授时响应中增加当前授时节点的状态，以使得各节点在接收到授时响应时，根据当前授时节点的状态调整本节点对授时节点发送授时请求的频率。比如，可以将授时节点的状态分为正常、空闲和繁忙三种类型。当任一节点获取到当前授时节点状态为空闲状态时，该任一节点可以针对当前所有待授时事务挨个向授时节点发送授时请求，从而提高对授时节点发送授时请求的频率，实现对授时

节点的充分利用，避免资源浪费，并且每个待授时事务都能获取到由授时节点分配的时间戳，充分保证了数据库的外部一致性。当任一节点获取到当前授时节点状态为繁忙状态时，可以先查询本节点上是否有符合前述实施例的其他事务，若有，则将其其他事务的时间戳分配至待授时事务，从而降低向授时节点发送获取时间戳请求的频率，减小授时节点的负载压力。当任一节点获取到当前授时节点状态为正常状态时，表明当前本节点向授时节点发送授时请求的频率处于平衡状态，只要保持当前频率就不会对授时节点造成过大的负载压力，同时还能保证授时节点处理授时请求的效率，避免返回授时响应过久的问题。

[44]与上述方法实施例相对应，本说明书还提供了一种装置的实施例。

[45]图4是根据本说明书一示例性实施例示出的一种电子设备的结构示意图。参考图4，在硬件层面，该电子设备包括处理器402、内部总线404、网络接口406、内存408以及非易失性存储器410，当然还可能包括其他业务所需要的硬件。处理器402从非易失性存储器410中读取对应的计算机程序到内存408中然后运行。当然，除了软件实现方式之外，本说明书并不排除其他实现方式，比如逻辑器件抑或软硬件结合的方式等等，也就是说以下处理流程的执行主体并不限于各个逻辑单元，也可以是硬件或逻辑器件。

[46]图5是根据本说明书一示例性实施例示出的一种应用于分布式数据库的时间戳分配装置的框图。参照图5，该装置包括提交模块502、查询模块504、分配模块506，其中：

[47]提交模块502，被配置为响应于接收的事务提交请求，提交相应的待授时事务并记录所述待授时事务的开始提交时刻。

[48]查询模块504，被配置为查询所述任一节点针对其他事务向所述授时节点发送授时请求的授时请求发送时刻。

[49]分配模块506，被配置为在任一其他事务的授时请求发送时刻不早于所述待授时事务的开始提交时刻的情况下，将所述授时节点针对该任一其他事务反馈的时间戳分配至所述待授时事务。

[50]可选的，所述查询模块504具体用于：查询所述任一节点针对最新已授时事务向所述授时节点发送授时请求的授时请求发送时刻；其中，在查询到的授时请求发送时刻不早于所述待授时事务的开始提交时刻的情况下，将所述最新已授时事务确定为所述任一其他事务。

[51]可选的，所述查询模块504还用于：在所述最新已授时事务的授时请求发送时刻早

于所述待授时事务的开始提交时刻的情况下，查询所述任一节点针对其他未授时事务向所述授时节点发送授时请求的授时请求发送时刻；其中，在任一其他未授时事务的授时请求发送时刻不早于所述待授时事务的开始提交时刻的情况下，将所述任一其他未授时事务作为所述任一其他事务。

[52]可选的，所述装置还包括：发送模块 508，被配置为在所有其他事务的授时请求发送时刻均早于所述待授时事务开始提交时刻的情况下，针对所述待授时事务向所述授时节点发送授时请求，以由所述授时节点响应于该授时请求向所述待授时事务分配时间戳。

[53]接收模块 510，被配置为接收所述授时节点返回的任一授时响应。

[54]更新模块 512，被配置为在所述任一授时响应对应的授时请求发送时刻晚于所述缓存中最新已授时事务的授时请求发送时刻的情况下，将所述缓存中最新已授时事务的授时请求发送时刻更新为所述授时响应对应的授时请求发送时刻。

[55]遍历模块 514，被配置为遍历所述缓存中待授时队列的所有待授时事务，将所述缓存中最新已授时事务的时间戳分配至所有满足条件的待授时事务，所述条件包括：所述最新已授时事务的授时请求发送时刻不早于所述待授时事务的开始提交时刻。

[56]调整模块 516，被配置为根据任一授时响应中包含的授时节点的当前运行状态调整所述任一节点对所述授时节点发送授时请求的频率，其中，所述授时节点的当前运行状态表明所述授时节点的繁忙程度。

[57]上述装置中各个模块的功能和作用的实现过程具体详见上述方法中对应步骤的实现过程，在此不再赘述。

[58]上述实施例阐明的装置或模块，具体可以由计算机芯片或实体实现，或者由具有某种功能的产品来实现。一种典型的实现设备为计算机，计算机的具体形式可以是个人计算机、膝上型计算机、蜂窝电话、相机电话、智能电话、个人数字助理、媒体播放器、导航设备、电子邮件收发设备、游戏控制台、平板计算机、可穿戴设备或者这些设备中的任意几种设备的组合。

[59]在一个典型的配置中，计算机包括一个或多个处理器 (CPU)、输入/输出接口、网络接口和内存。

[60]内存可能包括计算机可读介质中的非永久性存储器，随机存取存储器 (RAM) 和/或非易失性内存等形式，如只读存储器 (ROM) 或闪存(flash RAM)。内存是计算机可读介质的示例。

[61]计算机可读介质包括永久性和非永久性、可移动和非可移动媒体可以由任何方法或技术来实现信息存储。信息可以是计算机可读指令、数据结构、程序的模块或其他数据。计算机的存储介质的例子包括，但不限于相变内存 (PRAM)、静态随机存取存储器 (SRAM)、动态随机存取存储器 (DRAM)、其他类型的随机存取存储器 (RAM)、只读存储器 (ROM)、电可擦除可编程只读存储器 (EEPROM)、快闪记忆体或其他内存技术、只读光盘只读存储器 (CD-ROM)、数字多功能光盘 (DVD) 或其他光学存储、磁盒式磁带、磁盘存储、量子存储器、基于石墨烯的存储介质或其他磁性存储设备或任何其他非传输介质，可用于存储可以被计算设备访问的信息。按照本文中的界定，计算机可读介质不包括暂存电脑可读媒体 (transitory media)，如调制的数据信号和载波。

[62]以上所述仅为本说明书的较佳实施例而已，并不用以限制本说明书，凡在本说明书的精神和原则之内，所做的任何修改、等同替换、改进等，均应包含在本说明书保护的范围之内。

权利要求书

1、一种应用于分布式数据库的时间戳分配方法，所述分布式数据库的节点包括授时节点和普通节点，所述方法应用于任一节点，该方法包括：

响应于接收的事务提交请求，提交相应的待授时事务并记录所述待授时事务的开始提交时刻；

查询所述任一节点针对其他事务向所述授时节点发送授时请求的授时请求发送时刻；

在任一其他事务的授时请求发送时刻不早于所述待授时事务的开始提交时刻的情况下，将所述授时节点针对该任一其他事务反馈的时间戳分配至所述待授时事务。

2、根据权利要求 1 所述的方法，所述查询所述任一节点针对其他事务向所述授时节点发送授时请求的授时请求发送时刻，包括：

查询所述任一节点针对最新已授时事务向所述授时节点发送授时请求的授时请求发送时刻；

其中，在查询到的授时请求发送时刻不早于所述待授时事务的开始提交时刻的情况下，将所述最新已授时事务确定为所述任一其他事务。

3、根据权利要求 2 所述的方法，所述查询所述任一节点针对其他事务向所述授时节点发送授时请求的授时请求发送时刻，包括：

在所述最新已授时事务的授时请求发送时刻早于所述待授时事务的开始提交时刻的情况下，查询所述任一节点针对其他未授时事务向所述授时节点发送授时请求的授时请求发送时刻；

其中，在任一其他未授时事务的授时请求发送时刻不早于所述待授时事务的开始提交时刻的情况下，将所述任一其他未授时事务作为所述任一其他事务。

4、根据权利要求 1 所述的方法，还包括：

在所有其他事务的授时请求发送时刻均早于所述待授时事务的开始提交时刻的情况下，针对所述待授时事务向所述授时节点发送授时请求，以由所述授时节点响应于该授时请求向所述待授时事务分配时间戳。

5、根据权利要求 1 所述的方法，还包括：

接收所述授时节点返回的任一授时响应；

在所述任一授时响应对应的授时请求发送时刻晚于缓存中最新已授时事务的授时请求发送时刻的情况下，将所述缓存中最新已授时事务的授时请求发送时刻更新为所述任一授时响应对应的授时请求发送时刻。

6、根据权利要求1所述的方法，还包括：

接收所述授时节点返回的任一授时响应，所述任一授时响应包含授时节点的当前运行状态，所述授时节点的当前运行状态表明所述授时节点的繁忙程度；

根据所述授时节点的当前运行状态调整所述任一节点对所述授时节点发送授时请求的频率。

7、根据权利要求1所述的方法，还包括：

遍历缓存中待授时队列的所有待授时事务，将所述缓存中最新已授时事务的时间戳分配至所有满足条件的待授时事务，所述条件包括：所述最新已授时事务的授时请求发送时刻不早于所述待授时事务的开始提交时刻。

8、一种应用于分布式数据库的时间戳分配装置，所述分布式数据库的节点包括授时节点和普通节点，所述装置应用于任一节点，该装置包括：

提交模块，用于响应于接收的事务提交请求，提交相应的待授时事务并记录所述待授时事务的开始提交时刻；

查询模块，用于查询所述任一节点针对其他事务向所述授时节点发送授时请求的授时请求发送时刻；

分配模块，用于在任一其他事务的授时请求发送时刻不早于所述待授时事务的开始提交时刻的情况下，将所述授时节点针对该任一其他事务反馈的时间戳分配至所述待授时事务。

9、一种电子设备，包括：

处理器；

用于存储处理器可执行指令的存储器；

其中，所述处理器通过运行所述可执行指令以实现如权利要求1-7中任一项所述的方法。

10、一种计算机可读存储介质，其上存储有计算机指令，该指令被处理器执行时实现如权利要求1-7中任一项所述方法。