



(12) 发明专利

(10) 授权公告号 CN 110888739 B

(45) 授权公告日 2020.10.20

(21) 申请号 201911152333.5

(22) 申请日 2019.11.22

(65) 同一申请的已公布的文献号
申请公布号 CN 110888739 A

(43) 申请公布日 2020.03.17

(73) 专利权人 南京甄视智能科技有限公司
地址 211000 江苏省南京市江宁区高新园
龙眠大道568号

(72) 发明人 杨帆 张东冬 曹赛男

(74) 专利代理机构 南京行高知识产权代理有限
公司 32404

代理人 王培松

(51) Int. Cl.

G06F 9/50 (2006.01)

(56) 对比文件

CN 109582466 A, 2019.04.05

CN 108710537 A, 2018.10.26

CN 103744724 A, 2014.04.23

知乎. “优雅实现延时任务之Redis篇”.

《https://zhuanlan.zhihu.com/p/44914850》
.2018,

知乎. “优雅实现延时任务之Redis篇”.

《https://zhuanlan.zhihu.com/p/44914850》
.2018,

审查员 原野

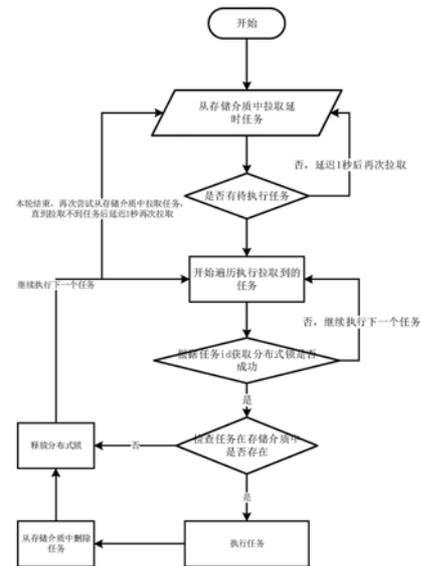
权利要求书2页 说明书6页 附图4页

(54) 发明名称

延迟任务的分布式处理方法与装置

(57) 摘要

本发明提供一种延迟任务的分布式处理方法与装置,包括:任务生产者确定任务预期执行时间戳;将任务id、预期执行时间戳、任务所需参数,合并为一个数据包存储到存储介质中;根据任务预期执行时间戳,将任务插入到执行任务队列的对应位置,然后将存储介质中的任务升序排列;集群内的各个节点作为任务消费者在周期内从存储介质中拉取预期时间戳小于当前时间的任务并且根据对任务的执行级别要求执行相应的处理。本发明利用分布式锁或者存储介质不会重复删除任务的特性,确保单个任务不会被重复执行,降低重复执行任务带来的风险。在任务生成时即确定了在存储介质中的位置,后续可快速拉取到待执行的任务,提高拉取和存储效率。



CN 110888739 B

1. 一种延迟任务的分布式处理方法,其特征在于,包括:
基于当前时间戳和延迟毫秒数确定任务预期执行时间戳;
将任务id、预期执行时间戳、任务所需参数,合并为一个数据包存储到存储介质中;
根据任务预期执行时间戳,将任务插入到执行任务队列的对应位置,然后将存储介质中的任务升序排列;

集群内的各个节点作为任务消费者在周期内从存储介质中拉取预期时间戳小于当前时间的任务,每次拉取的任务数量小于预设数量;如果没有拉取到符合条件的任务,那么本次拉取结束,等待周期时间后再次拉取;如果拉取到任务,则根据对任务的执行级别要求执行,其中执行级别包括保证任务必须被执行以及任务不需要保证必须被执行;

其中:

1) 在执行任务的过程中,对于保证任务必须被执行的要求,集群内的各个节点的执行处理包括:

根据任务id尝试获取分布式锁,如果锁获取失败,则判定当前任务正在被其他节点执行,跳过该任务,执行下一个任务;

如果分布式锁获取成功,则请求存储介质检查当前任务是否存在;

如果在存储介质中检查到当前任务依然存在,则判定任务还未被其他节点执行,并执行任务,执行结束从存储介质中删除该任务,同时释放分布式锁,继续执行下一个任务;

如果存储介质中检查不到当前任务,则判定任务已被其他节点执行,那么释放分布式锁,继续执行下一个任务;

2) 在执行任务的过程中,对于任务不需要保证必须被执行的要求,集群内的各个节点的执行处理包括:跳过分式锁,通过判断任务是否成功从存储介质中移除来确保任务被首次执行。

2. 根据权利要求1所述的延迟任务的分布式处理方法,其特征在于,在执行任务的过程中,对于保证任务必须被执行和任务不需要保证必须被执行具有对应的标记以进行标记和识别。

3. 根据权利要求1所述的延迟任务的分布式处理方法,其特征在于,在拉取到任务后,首先尝试从存储介质中移除该任务;如果从存储介质中移除成功,则判定该任务还没有被其他节点执行,继续执行任务;如果从存储介质中移除失败,则判定任务已经被其他节点执行,跳过该任务,执行下一个任务。

4. 根据权利要求1所述的延迟任务的分布式处理方法,其特征在于,所述任务预期执行时间戳的确定方式为:当前时间戳+延迟毫秒数。

5. 根据权利要求1所述的延迟任务的分布式处理方法,其特征在于,在拉去任务过程中,基于预设的拉取频率拉取。

6. 一种延迟任务的分布式处理装置,其特征在于,包括:

用于当前时间戳和延迟毫秒数确定任务预期执行时间戳的模块;

用于将任务id、预期执行时间戳、任务所需参数,合并为一个数据包存储到存储介质中的模块;

用于根据任务预期执行时间戳,将任务插入到执行任务队列的对应位置,然后将存储介质中的任务升序排列的模块;

用于使集群内的各个节点作为任务消费者在周期内从存储介质中拉取预期时间戳小于当前时间的任务并且根据对任务的执行级别要求执行相应的处理的模块,其中,每次拉取的任务数量小于预设数量;如果没有拉取到符合条件的任务,则本次拉取结束,等待周期时间后再次拉取;如果拉取到任务,则根据对任务的执行级别要求执行,其中执行级别包括保证任务必须被执行以及任务不需要保证必须被执行。

7. 根据权利要求6所述的延迟任务的分布式处理装置,其特征在于,还包括:

第一执行模块,用于对于保证任务必须被执行的要求的执行处理;以及

第二执行模块,用于对于任务不需要保证必须被执行的要求的执行处理。

8. 根据权利要求7所述的延迟任务的分布式处理装置,其特征在于,所述第一执行模块的处理操作包括:

根据任务id尝试获取分布式锁,如果锁获取失败,则判定当前任务正在被其他节点执行,跳过该任务,执行下一个任务;

如果分布式锁获取成功,则请求存储介质检查当前任务是否存在;

如果在存储介质中检查到当前任务依然存在,则判定任务还未被其他节点执行,并执行任务,执行结束从存储介质中删除该任务,同时释放分布式锁,继续执行下一个任务;

如果存储介质中检查不到当前任务,则判定任务已被其他节点执行,那么释放分布式锁,继续执行下一个任务;

所述第二执行模块的执行处理包括:

通过判断任务是否成功从存储介质中移除来确保任务被首次执行,其中节点在拉取到任务后,首先尝试从存储介质中移除该任务;如果从存储介质中移除成功,则判定该任务还没有被其他节点执行,继续执行任务;如果从存储介质中移除失败,则判定任务已经被其他节点执行,跳过该任务,执行下一个任务。

延迟任务的分布式处理方法与装置

技术领域

[0001] 本发明涉及计算机系统领域,具体而言涉及一种延迟任务的分布式处理方法与装置。

背景技术

[0002] 在应用开发过程中,经常会遇到需要在指定时间到来时去执行相应延时任务的业务需求。例如,订单提交之后,若超过指定支付时间未支付成功,则需要将该订单删除,再例如,订单完成之后,若超过指定评价时间未及时评价,则默认将该订单的评价设置为五星好评,又例如,代金券等虚拟网络资源在被激活之后,若超过指定使用时间未被使用,则自动作废。

[0003] 目前,针对此类业务需求,在一些实施方式中,会在接收到延时任务之后,将延时任务的触发信息存储于数据库,同时,通过定时任务周期性的去扫描该数据库,以检测是否存在需要执行的延时任务,若存在需要执行的延时任务,则一次性批量执行,图1是现有的延迟任务的处理流程,包括:向数据库插入需要延迟执行任务的时间点和任务参数信息;按照任务触发时间排序查询数据库待执行的任务;执行满足时间点的任务;更新任务执行状态。但在实际处理过程中,我们发现这种方法存在以下问题:单个节点可能执行了过多任务,会导致排序在后的延时任务的执行时间滞后,而其他节点的资源没有被有效利用。

[0004] 而数据库属于底层公共设施,查询和存储效率会受其他应用影响,例如某个服务阻塞了数据库查询连接,等到可以查询任务时,已经过了比较长的时间,导致有可能存在延迟任务的延迟时间大于预期的时间。

[0005] 同时,数据库在查询时需要任务按照规则进行排序,需要消耗额外的性能。如果并发执行任务,可能导致A节点执行了任务,但是还没有更新状态,此时B节点也拉取到了任务并执行成功,导致任务重复执行。

发明内容

[0006] 本发明目的在于提供一种与数据库解耦并避免与系统中其他应用争夺数据库资源的延迟任务的分布式处理方法与装置,可保证集群内的每一个节点都可以消费延迟任务,各个节点的压力均衡。

[0007] 为实现上述目的,本发明所采用的技术方案包括:

[0008] 一种延迟任务的分布式处理方法与,包括:

[0009] 基于当前时间戳和延迟毫秒数确定任务预期执行时间戳;

[0010] 将任务id、预期执行时间戳、任务所需参数,合并为一个数据包存储到存储介质中;

[0011] 根据任务预期执行时间戳,将任务插入到执行任务队列的对应位置,然后将存储介质中的任务升序排列;

[0012] 集群内的各个节点作为任务消费者在周期内从存储介质中拉取预期时间戳小于

当前时间的任务,每次拉取的任务数量小于预设数量;如果没有拉取到符合条件的任务,那么本次拉取结束,等待周期时间后再次拉取;如果拉取到任务,则根据对任务的执行级别要求执行,其中执行级别包括保证任务必须被执行以及任务不需要保证必须被执行。

[0013] 优选地,在执行任务的过程中,对于保证任务必须被执行和任务不需要保证必须被执行具有对应的标记以进行标记和识别。

[0014] 优选地,在执行任务的过程中,对于保证任务必须被执行的要求,集群内的各个节点的执行处理包括:

[0015] 根据任务id尝试获取分布式锁,如果锁获取失败,则判定当前任务正在被其他节点执行,跳过该任务,执行下一个任务;

[0016] 如果分布式锁获取成功,则请求存储介质检查当前任务是否存在;

[0017] 如果在存储介质中检查到当前任务依然存在,则判定任务还未被其他节点执行,并执行任务,执行结束从存储介质中删除该任务,同时释放分布式锁,继续执行下一个任务;

[0018] 如果存储介质中检查不到当前任务,则判定任务已被其他节点执行,那么释放分布式锁,继续执行下一个任务。

[0019] 优选地,在执行任务的过程中,对于任务不需要保证必须被执行的要求,集群内的各个节点的执行处理包括:

[0020] 跳过分布式锁,通过判断任务是否成功从存储介质中移除来确保任务被首次执行。

[0021] 优选地,在拉取到任务后,首先尝试从存储介质中移除该任务;如果从存储介质中移除成功,则判定该任务还没有被其他节点执行,继续执行任务;如果从存储介质中移除失败,则判定任务已经被其他节点执行,跳过该任务,执行下一个任务。

[0022] 优选地,任务预期执行时间戳的确定方式为:当前时间戳+延迟毫秒数。

[0023] 优选地,,在拉去任务过程中,基于预设的拉取频率拉取。

[0024] 根据本发明的改进,还提出一种延迟任务的分布式处理装置,包括:

[0025] 用于当前时间戳和延迟毫秒数确定任务预期执行时间戳的模块;

[0026] 用于将任务id、预期执行时间戳、任务所需参数,合并为一个数据包存储到存储介质中的模块;

[0027] 用于根据任务预期执行时间戳,将任务插入到执行任务队列的对应位置,然后将存储介质中的任务升序排列的模块;

[0028] 用于使集群内的各个节点作为任务消费者在周期内从存储介质中拉取预期时间戳小于当前时间的任务并且根据对任务的执行级别要求执行相应的处理的模块,其中,每次拉取的任务数量小于预设数量;如果没有拉取到符合条件的任务,则本次拉取结束,等待周期时间后再次拉取;如果拉取到任务,则根据对任务的执行级别要求执行,其中执行级别包括保证任务必须被执行以及任务不需要保证必须被执行。

[0029] 优选地,还包括:

[0030] 第一执行模块,用于对于保证任务必须被执行的要求的执行处理;以及

[0031] 第二执行模块,用于对于任务不需要保证必须被执行的要求的执行处理。

[0032] 优选地,所述第一执行模块的处理操作包括:

[0033] 根据任务id尝试获取分布式锁,如果锁获取失败,则判定当前任务正在被其他节点执行,跳过该任务,执行下一个任务;

[0034] 如果分布式锁获取成功,则请求存储介质检查当前任务是否存在;

[0035] 如果在存储介质中检查到当前任务依然存在,则判定任务还未被其他节点执行,并执行任务,执行结束从存储介质中删除该任务,同时释放分布式锁,继续执行下一个任务;

[0036] 如果存储介质中检查不到当前任务,则判定任务已被其他节点执行,那么释放分布式锁,继续执行下一个任务;

[0037] 所述第二执行模块的执行处理包括:

[0038] 通过判断任务是否成功从存储介质中移除来确保任务被首次执行,其中节点在拉取到任务后,首先尝试从存储介质中移除该任务;如果从存储介质中移除成功,则判定该任务还没有被其他节点执行,继续执行任务;如果从存储介质中移除失败,则判定任务已经被其他节点执行,跳过该任务,执行下一个任务。

[0039] 应当理解,前述构思以及在下面更加详细地描述的额外构思的所有组合只要在这样的构思不相互矛盾的情况下都可以被视为本公开的发明主题的一部分。另外,所要求保护的的主题的所有组合都被视为本公开的发明主题的一部分。

[0040] 结合附图从下面的描述中可以更加全面地理解本发明教导的前述和其他方面、实施例和特征。本发明的其他附加方面例如示例性实施方式的特征和/或有益效果将在下面的描述中显见,或通过根据本发明教导的具体实施方式的实践中得知。

附图说明

[0041] 附图不意在按比例绘制。在附图中,在各个图中示出的每个相同或近似相同的组成部分可以用相同的标号表示。为了清晰起见,在每个图中,并非每个组成部分均被标记。现在,将通过例子并参考附图来描述本发明的各个方面的实施例,其中:

[0042] 图1是现有技术中延迟任务的处理流程示意图。

[0043] 图2是本发明示例性实施例的延迟任务分布式处理方法的任务提交流程示例。

[0044] 图3是本发明示例性实施例的延迟任务分布式处理方法的第一任务消费处理的流程示意图。

[0045] 图4是本发明示例性实施例的延迟任务分布式处理方法的第二任务消费处理的流程示意图。

具体实施方式

[0046] 为了更了解本发明的技术内容,特举具体实施例并配合所附图式说明如下。

[0047] 在本公开中参照附图来描述本发明的各方面,附图中示出了许多说明的实施例。本公开的实施例不必定意在包括本发明的所有方面。应当理解,上面介绍的多种构思和实施例,以及下面更加详细地描述的那些构思和实施方式可以以很多方式中任意一种来实施,这是因为本发明所公开的构思和实施例并不限于任何实施方式。另外,本发明公开的一些方面可以单独使用,或者与本发明公开的其他方面的任何适当组合来使用。

[0048] 本发明旨在针对现有的延迟任务处理对数据库的依赖问题,导致受其他应用高峰

期操作数据库导致的数据库性能下降,影响延迟任务的存储和查询的问题,尤其是还会出现任务的重复执行和无节点执行的问题,提出一种延迟任务的分布式处理方法与,结合图2-4所示,其实现过程包括:

[0049] 基于当前时间戳和延迟毫秒数确定任务预期执行时间戳;

[0050] 将任务id、预期执行时间戳、任务所需参数,合并为一个数据包存储到存储介质中;

[0051] 根据任务预期执行时间戳,将任务插入到执行任务队列的对应位置,然后将存储介质中的任务升序排列;

[0052] 集群内的各个节点作为任务消费者在周期内从存储介质中拉取预期时间戳小于当前时间的任务,每次拉取的任务数量小于预设数量;如果没有拉取到符合条件的任务,那么本次拉取结束,等待周期时间后再次拉取;如果拉取到任务,则根据对任务的执行级别要求执行,其中执行级别包括保证任务必须被执行以及任务不需要保证必须被执行。

[0053] 优选地,在执行任务的过程中,对于保证任务必须被执行和任务不需要保证必须被执行具有对应的标记以进行标记和识别。

[0054] 优选地,在执行任务的过程中,对于保证任务必须被执行的要求,集群内的各个节点的执行处理包括:

[0055] 根据任务id尝试获取分布式锁,如果锁获取失败,则判定当前任务正在被其他节点执行,跳过该任务,执行下一个任务;

[0056] 如果分布式锁获取成功,则请求存储介质检查当前任务是否存在;

[0057] 如果在存储介质中检查到当前任务依然存在,则判定任务还未被其他节点执行,并执行任务,执行结束从存储介质中删除该任务,同时释放分布式锁,继续执行下一个任务;

[0058] 如果存储介质中检查不到当前任务,则判定任务已被其他节点执行,那么释放分布式锁,继续执行下一个任务。

[0059] 优选地,在执行任务的过程中,对于任务不需要保证必须被执行的要求,集群内的各个节点的执行处理包括:

[0060] 跳过分布式锁,通过判断任务是否成功从存储介质中移除来确保任务被首次执行。

[0061] 优选地,在拉取到任务后,首先尝试从存储介质中移除该任务;如果从存储介质中移除成功,则判定该任务还没有被其他节点执行,继续执行任务;如果从存储介质中移除失败,则判定任务已经被其他节点执行,跳过该任务,执行下一个任务。

[0062] 优选地,任务预期执行时间戳的确定方式为:当前时间戳+延迟毫秒数。

[0063] 优选地,,在拉去任务过程中,基于预设的拉取频率拉取。

[0064] 下面结合一些示例对前述过程进行更加具体的说明。

[0065] 本发明的示例中,毫秒表示时间单位,1秒=1000毫秒。

[0066] 时间戳,表示例如从1970-01-01 08:00:00到某一个时间点的毫秒数,例如1571131426353表示的时间点为2019-10-15 17:23:46。

[0067] 任务生产者计算出任务预期执行时间戳,获取方式为当前时间戳+延迟毫秒数,例如,当前时间戳为1570863741545,延迟执行时间为10分钟,那么计算结果为1570863741545

+10*60*1000=1570863741545。

[0068] 群内的各个节点作为任务消费者(以下简称消费者)在周期内从存储介质中拉取预期时间戳小于当前时间的任务,拉取频率可以设置为1秒/次或者500毫秒/次,每次拉取的任务数量应当小于指定数量,例如10个。

[0069] 优选地,本发明各个实施例中的分布式锁可以根据本领域的需要选用,本案例使用Redisson作为分布式锁实现方案。

[0070] 如此,本发明的延迟任务的分布式处理,可实现任务消费与数据库解耦,避免与系统中其他应用争夺数据库资源,而集群内的每一个节点都可以消费延迟任务,各个节点的压力均衡,避免一个多个节点故障导致任务无法被继续执行。

[0071] 本发明利用分布式锁或者存储介质不会重复删除任务的特性,确保单个任务不会被重复执行,降低重复执行任务带来的业务风险。同时,优化任务存储结构,在任务生成时即确定了在存储介质中的位置,后续可以快速拉取到待执行的任务,提高拉取和存储效率。

[0072] 结合图示,本发明的前述实施例还可以根据具体实施配置如下。

[0073] {分布式处理装置}

[0074] 还提出一种延迟任务的分布式处理装置,包括:

[0075] 用于当前时间戳和延迟毫秒数确定任务预期执行时间戳的模块;

[0076] 用于将任务id、预期执行时间戳、任务所需参数,合并为一个数据包存储到存储介质中的模块;

[0077] 用于根据任务预期执行时间戳,将任务插入到执行任务队列的对应位置,然后将存储介质中的任务升序排列的模块;

[0078] 用于使集群内的各个节点作为任务消费者在周期内从存储介质中拉取预期时间戳小于当前时间的任务并且根据对任务的执行级别要求执行相应的处理的模块,其中,每次拉取的任务数量小于预设数量;如果没有拉取到符合条件的任务,则本次拉取结束,等待周期时间后再次拉取;如果拉取到任务,则根据对任务的执行级别要求执行,其中执行级别包括保证任务必须被执行以及任务不需要保证必须被执行。

[0079] 优选地,还包括:

[0080] 第一执行模块,用于对于保证任务必须被执行的要求的执行处理;以及

[0081] 第二执行模块,用于对于任务不需要保证必须被执行的要求的执行处理。

[0082] 优选地,所述第一执行模块的处理操作包括:

[0083] 根据任务id尝试获取分布式锁,如果锁获取失败,则判定当前任务正在被其他节点执行,跳过该任务,执行下一个任务;

[0084] 如果分布式锁获取成功,则请求存储介质检查当前任务是否存在;

[0085] 如果在存储介质中检查到当前任务依然存在,则判定任务还未被其他节点执行,并执行任务,执行结束从存储介质中删除该任务,同时释放分布式锁,继续执行下一个任务;

[0086] 如果存储介质中检查不到当前任务,则判定任务已被其他节点执行,那么释放分布式锁,继续执行下一个任务;

[0087] 所述第二执行模块的执行处理包括:

[0088] 通过判断任务是否成功从存储介质中移除来确保任务被首次执行,其中节点在拉

取到任务后,首先尝试从存储介质中移除该任务;如果从存储介质中移除成功,则判定该任务还没有被其他节点执行,继续执行任务;如果从存储介质中移除失败,则判定任务已经被其他节点执行,跳过该任务,执行下一个任务。

[0089] 虽然本发明已以较佳实施例揭露如上,然其并非用以限定本发明。本发明所属技术领域中具有通常知识者,在不脱离本发明的精神和范围内,当可作各种的更动与润饰。因此,本发明的保护范围当视权利要求书所界定者为准。

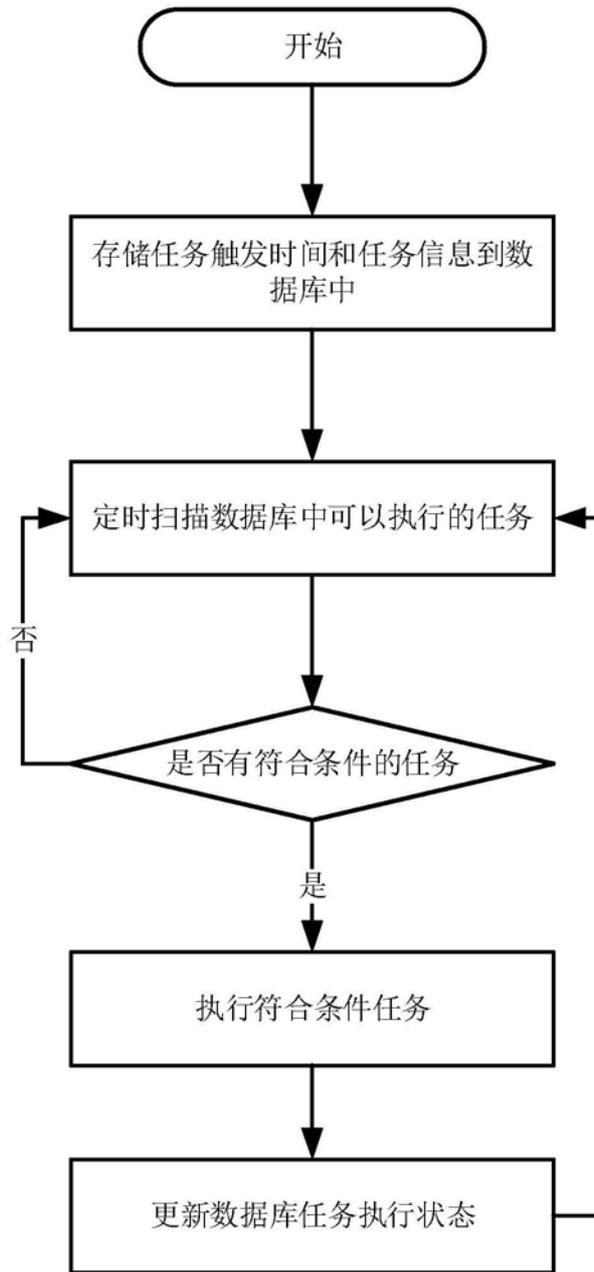


图1

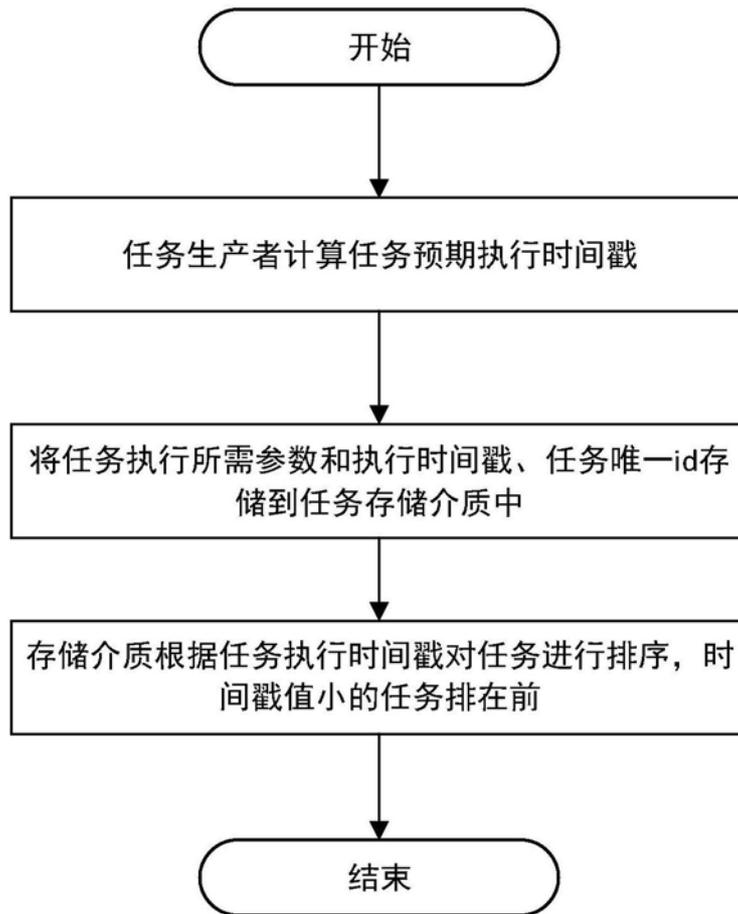


图2

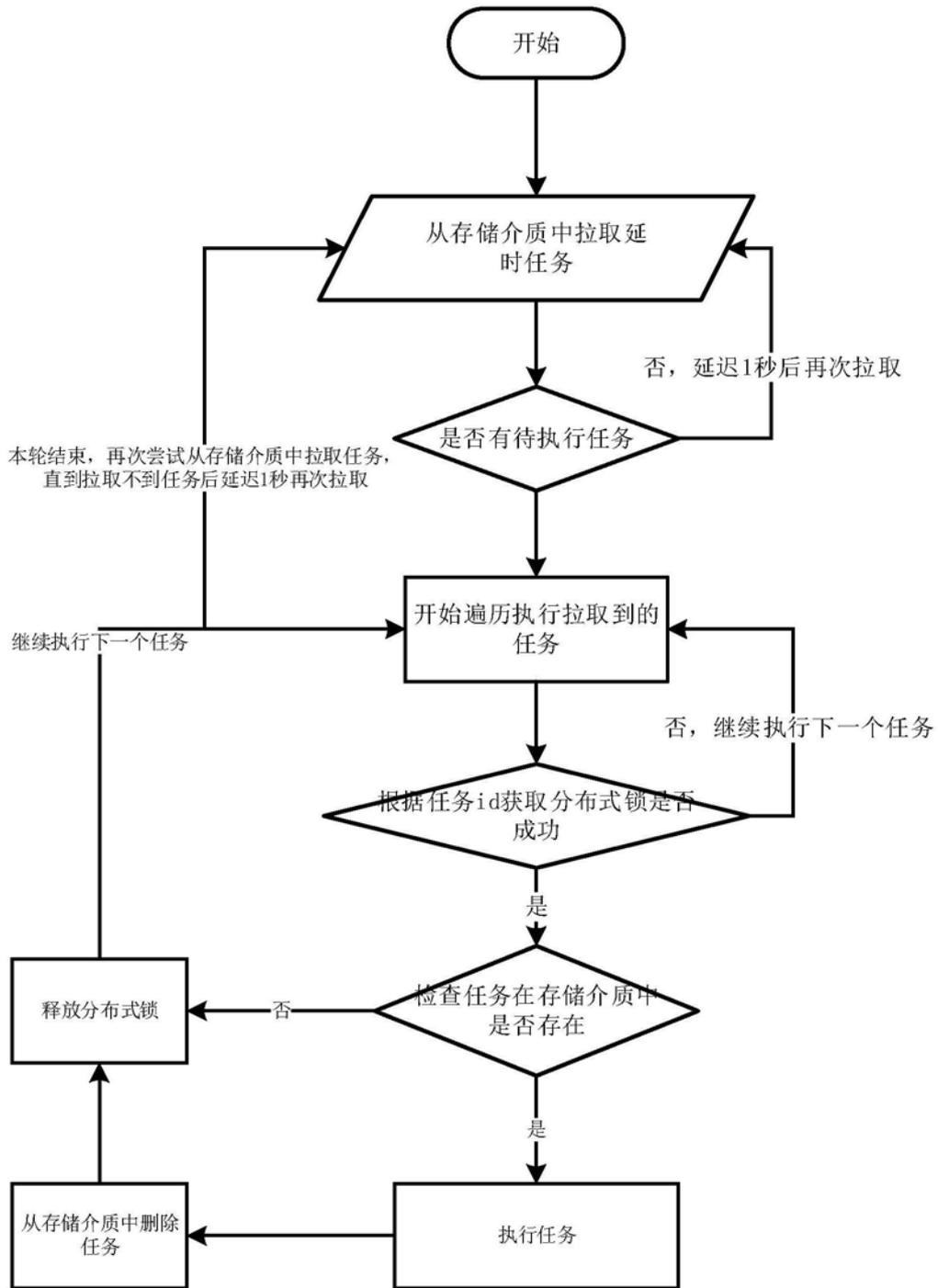


图3

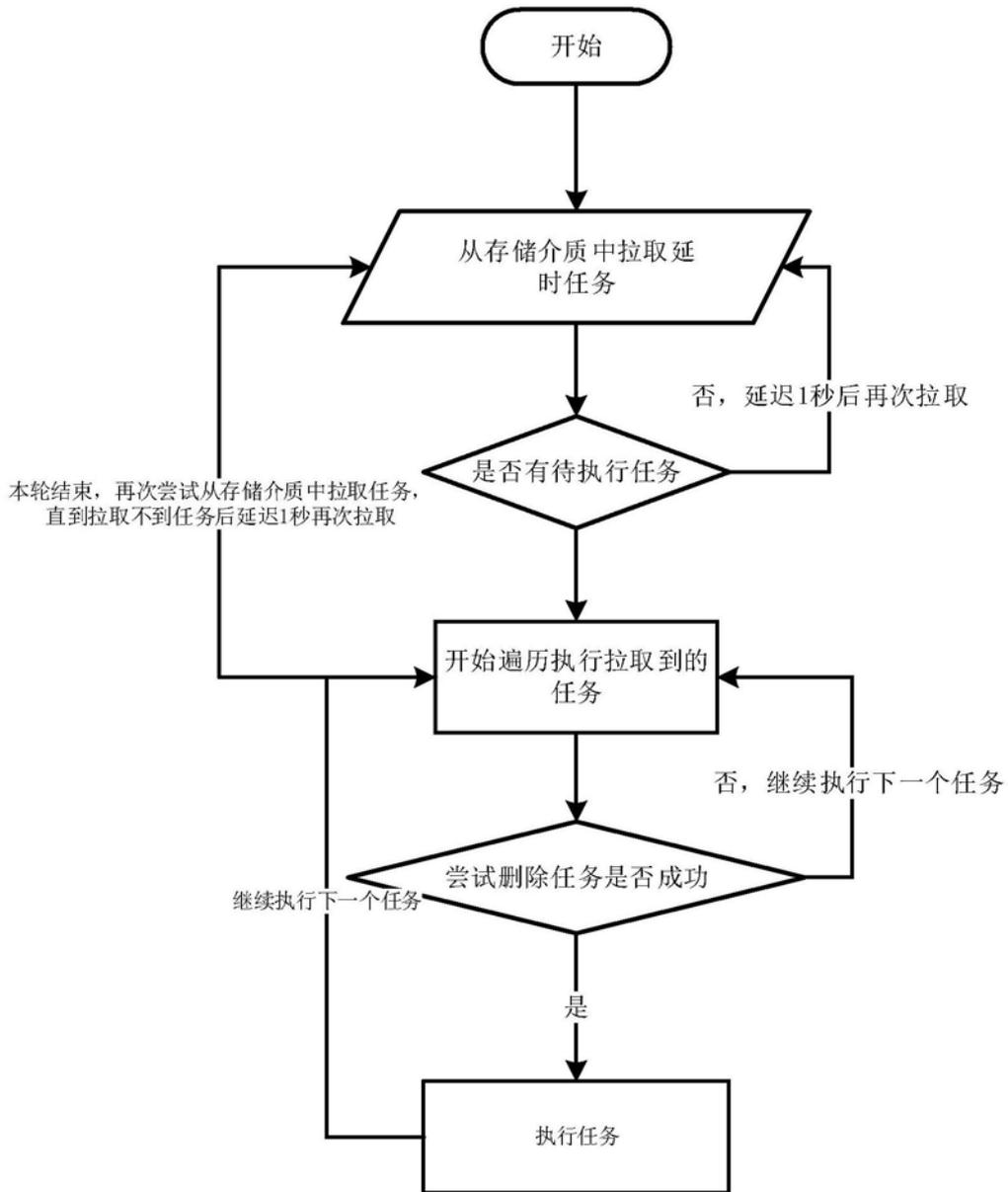


图4