



(12) 发明专利

(10) 授权公告号 CN 111813552 B

(45) 授权公告日 2021.12.03

(21) 申请号 202010686320.2

G06F 9/54 (2006.01)

(22) 申请日 2020.07.16

(56) 对比文件

(65) 同一申请的已公布的文献号

CN 101599027 A, 2009.12.09

申请公布号 CN 111813552 A

CN 101727352 A, 2010.06.09

(43) 申请公布日 2020.10.23

CN 110134500 A, 2019.08.16

(73) 专利权人 济南浪潮数据技术有限公司

CN 108536530 A, 2018.09.14

地址 250101 山东省济南市高新区浪潮路
1036号浪潮科技园S05楼S311室

US 2004064580 A1, 2004.04.01

(72) 发明人 张珠玉

US 2009235099 A1, 2009.09.17

(74) 专利代理机构 北京集佳知识产权代理有限公司 11227

US 2018349177 A1, 2018.12.06

代理人 史翠

US 2002059503 A1, 2002.05.16

(51) Int.Cl.

王嘉等.一种改进型多线程机制在数据采集
系统中的应用.《计算技术与自动化》.2019,第38
卷(第4期),

G06F 9/50 (2006.01)

审查员 鱼冰

G06F 9/48 (2006.01)

权利要求书2页 说明书9页 附图2页

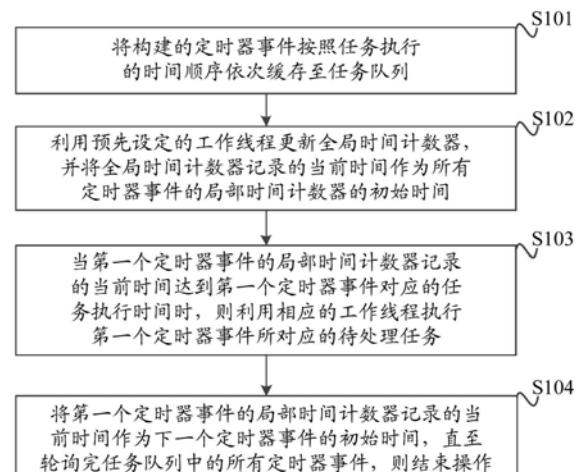
(54) 发明名称

一种基于多线程任务的调度执行方法、装置
和介质

(57) 摘要

本发明实施例公开了一种基于多线程任务的调度执行方法、装置和介质，将构建的定时器事件按照任务执行的时间顺序依次缓存至任务队列；利用预先设定的工作线程更新全局时间计数器，并将全局时间计数器记录的当前时间作为所有局部时间计数器的初始时间。当第一个定时器事件的局部时间计数器记录的当前时间达到第一个定时器事件对应的任务执行时间时，则利用相应的工作线程执行第一个定时器事件所对应的待处理任务；将第一个定时器事件的局部时间计数器记录的当前时间作为下一个定时器事件的初始时间，直至轮询完任务队列中的所有定时器事件，则结束操作。在不增加成本的前提下，解决了多线程任务调度执行过程中的时钟同步问题。

B
CN 111813552



CN

1. 一种基于多线程任务的调度执行方法,其特征在于,包括:

将构建的定时器事件按照任务执行的时间顺序依次缓存至任务队列;其中,每个定时器事件有其对应的一个局部时间计数器以及工作线程;

利用预先设定的工作线程更新全局时间计数器,并将所述全局时间计数器记录的当前时间作为所有所述定时器事件的局部时间计数器的初始时间;

当第一个定时器事件的局部时间计数器记录的当前时间达到所述第一个定时器事件对应的任务执行时间时,则利用相应的工作线程执行所述第一个定时器事件所对应的待处理任务;

将所述第一个定时器事件的局部时间计数器记录的当前时间作为下一个定时器事件的初始时间,直至轮询完所述任务队列中的所有定时器事件,则结束操作。

2. 根据权利要求1所述的基于多线程任务的调度执行方法,其特征在于,所述将构建的定时器事件按照任务执行的时间顺序依次缓存至任务队列包括:

基于待处理任务构建定时器事件;其中,每个定时器事件包括前后节点指向指针、触发时刻、迭代周期和增量触发时间;

按照各定时器事件的触发时刻将所有所述定时器事件升序排列缓存至任务队列。

3. 根据权利要求2所述的基于多线程任务的调度执行方法,其特征在于,在所述将构建的定时器事件按照任务执行的时间顺序依次缓存至任务队列之后还包括:

当接收到新的待处理任务时,对所述新的待处理任务构建新的定时器事件;

根据所述新的定时器事件的触发时刻,将所述新的定时器事件插入所述任务队列,并更新与所述新的定时器事件前后相邻的两个定时器事件的前后节点指向指针和增量触发时间。

4. 根据权利要求1所述的基于多线程任务的调度执行方法,其特征在于,所述利用预先设定的工作线程更新全局时间计数器包括:

按照锁保护的方式,利用多个高优先级工作线程更新全局时间计数器。

5. 根据权利要求1所述的基于多线程任务的调度执行方法,其特征在于,在所述利用相应的工作线程执行所述第一个定时器事件所对应的待处理任务之后还包括:

当完成所述第一个定时器事件所对应的待处理任务之后,将所述第一个定时器事件从所述任务队列中删除。

6. 根据权利要求1-5任意一项所述的基于多线程任务的调度执行方法,其特征在于,还包括:

计算各所述局部时间计数器记录的当前时间与所述全局时间计数器记录的当前时间的差值;

当存在差值大于预设值的目标局部时间计数器时,则对所述目标局部时间计数器所对应的工作进程进行报错提示。

7. 一种基于多线程任务的调度执行装置,其特征在于,包括缓存单元、更新单元、执行单元和作为单元;

所述缓存单元,用于将构建的定时器事件按照任务执行的时间顺序依次缓存至任务队列;其中,每个定时器事件有其对应的一个局部时间计数器以及工作线程;

所述更新单元,用于利用预先设定的工作线程更新全局时间计数器,并将所述全局时

间计数器记录的当前时间作为所有所述定时器事件的局部时间计数器的初始时间；

所述执行单元，用于当第一个定时器事件的局部时间计数器记录的当前时间达到所述第一个定时器事件对应的任务执行时间时，则利用相应的工作线程执行所述第一个定时器事件所对应的待处理任务；

所述作为单元，用于将所述第一个定时器事件的局部时间计数器记录的当前时间作为下一个定时器事件的初始时间，直至轮询完所述任务队列中的所有定时器事件，则结束操作。

8. 根据权利要求7所述的基于多线程任务的调度执行装置，其特征在于，缓存至缓存单元包括构建子单元和排列子单元；

所述构建子单元，用于基于待处理任务构建定时器事件；其中，每个定时器事件包括前后节点指向指针、触发时刻、迭代周期和增量触发时间；

所述排列子单元，用于按照各定时器事件的触发时刻将所有所述定时器事件升序排列缓存至任务队列。

9. 一种基于多线程任务的调度执行装置，其特征在于，包括：

存储器，用于存储计算机程序；

处理器，用于执行所述计算机程序以实现如权利要求1至6任意一项所述基于多线程任务的调度执行方法的步骤。

10. 一种计算机可读存储介质，其特征在于，所述计算机可读存储介质上存储有计算机程序，所述计算机程序被处理器执行时实现如权利要求1至6任意一项所述基于多线程任务的调度执行方法的步骤。

一种基于多线程任务的调度执行方法、装置和介质

技术领域

[0001] 本发明涉及存储系统技术领域,特别是涉及一种基于多线程任务的调度执行方法、装置和计算机可读存储介质。

背景技术

[0002] 随着存储系统要处理的IO请求(即所要执行的任务)越来越多,对存储系统的性能要求也越来越高。在所要执行的任务越来越多的情况下,为了提高任务处理性能,目前,常采用的方法是创建尽可能多的线程来增加并发度。多线程任务调度执行应用背景下,定时器处理模式的选择与设计则显得尤为关键。

[0003] 现有的计算机多节点系统中,普遍采用硬件和软件两种思路实现定时器功能。硬件方式可以包括卫星信号定时和电子器件脉冲定时,但是硬件方式需另外购进第三方设备合入系统,增加了建设成本,并且增加了系统搭建的复杂度。为了降低建设成本,以多媒体定时器为代表的软件定时器成为一个应用范围较大的思路。软件定时器依赖于操作系统,精确性高,但严重受时区和系统时钟修改影响。在性能要求稳定的存储系统领域,现有的软件定时器思路不是很好的选择。

[0004] 可见,如何在不增加成本的前提下,解决多线程任务调度执行过程中的时钟同步问题,是本领域技术人员需要解决的问题。

发明内容

[0005] 本发明实施例的目的是提供一种基于多线程任务的调度执行方法、装置和计算机可读存储介质,可以在不增加成本的前提下,解决多线程任务调度执行过程中的时钟同步问题。

[0006] 为解决上述技术问题,本发明实施例提供一种基于多线程任务的调度执行方法,包括:

[0007] 将构建的定时器事件按照任务执行的时间顺序依次缓存至任务队列;其中,每个定时器事件有其对应的一个局部时间计数器以及工作线程;

[0008] 利用预先设定的工作线程更新全局时间计数器,并将所述全局时间计数器记录的当前时间作为所有所述定时器事件的局部时间计数器的初始时间;

[0009] 当第一个定时器事件的局部时间计数器记录的当前时间达到所述第一个定时器事件对应的任务执行时间时,则利用相应的工作线程执行所述第一个定时器事件所对应的待处理任务;

[0010] 将所述第一个定时器事件的局部时间计数器记录的当前时间作为下一个定时器事件的初始时间,直至轮询完所述任务队列中的所有定时器事件,则结束操作。

[0011] 可选地,所述将构建的定时器事件按照任务执行的时间顺序依次缓存至任务队列包括:

[0012] 基于待处理任务构建定时器事件;其中,每个定时器事件包括前后节点指向指针、

触发时刻、迭代周期和增量触发时间；

[0013] 按照各定时器事件的触发时刻将所有所述定时器事件升序排列缓存至任务队列。

[0014] 可选地，在所述将构建的定时器事件按照任务执行的时间顺序依次缓存至任务队列之后还包括：

[0015] 当接收到新的待处理任务时，对所述新的待处理任务构建新的定时器事件；

[0016] 根据所述新的定时器事件的触发时刻，将所述新的定时器事件插入所述任务队列，并更新与所述新的定时器事件前后相邻的两个定时器事件的前后节点指向指针和增量触发时间。

[0017] 可选地，所述利用预先设定的工作线程更新全局时间计数器包括：

[0018] 按照锁保护的方式，利用多个高优先级工作线程更新全局时间计数器。

[0019] 可选地，在所述利用相应的工作线程执行所述第一个定时器事件所对应的待处理任务之后还包括：

[0020] 当完成所述第一个定时器事件所对应的待处理任务之后，将所述第一个定时器事件从所述任务队列中删除。

[0021] 可选地，还包括：

[0022] 计算各所述局部时间计数器记录的当前时间与所述全局时间计数器记录的当前时间的差值；

[0023] 当存在差值大于预设值的目标局部时间计数器时，则对所述目标局部时间计数器所对应的工作进程进行报错提示。

[0024] 本发明实施例还提供了一种基于多线程任务的调度执行装置，包括缓存单元、更新单元、执行单元和作为单元；

[0025] 所述存储单元，用于将构建的定时器事件按照任务执行的时间顺序依次缓存至任务队列；其中，每个定时器事件有其对应的一个局部时间计数器以及工作线程；

[0026] 所述更新单元，用于利用预先设定的工作线程更新全局时间计数器，并将所述全局时间计数器记录的当前时间作为所有所述定时器事件的局部时间计数器的初始时间；

[0027] 所述执行单元，用于当第一个定时器事件的局部时间计数器记录的当前时间达到所述第一个定时器事件对应的任务执行时间时，则利用相应的工作线程执行所述第一个定时器事件所对应的待处理任务；

[0028] 所述作为单元，用于将所述第一个定时器事件的局部时间计数器记录的当前时间作为下一个定时器事件的初始时间，直至轮询完所述任务队列中的所有定时器事件，则结束操作。

[0029] 可选地，所述缓存至缓存单元包括构建子单元和排列子单元；

[0030] 所述构建子单元，用于基于待处理任务构建定时器事件；其中，每个定时器事件包括前后节点指向指针、触发时刻、迭代周期和增量触发时间；

[0031] 所述排列子单元，用于按照各定时器事件的触发时刻将所有所述定时器事件升序排列缓存至任务队列。

[0032] 可选地，还包括构建单元、插入单元和指针更新单元；

[0033] 所述构建单元，用于当接收到新的待处理任务时，对所述新的待处理任务构建新的定时器事件；

[0034] 所述插入单元,用于根据所述新的定时器事件的触发时刻,将所述新的定时器事件插入所述任务队列;

[0035] 所述指针更新单元,用于更新与所述新的定时器事件前后相邻的两个定时器事件的前后节点指向指针和增量触发时间。

[0036] 可选地,所述更新单元包括按照锁保护的方式,利用多个高优先级工作线程更新全局时间计数器。

[0037] 可选地,还包括删除单元;

[0038] 所述删除单元,用于当完成所述第一个定时器事件所对应的待处理任务之后,将所述第一个定时器事件从所述任务队列中删除。

[0039] 可选地,还包括计算单元和提示单元;

[0040] 所述计算单元,用于计算各所述局部时间计数器记录的当前时间与所述全局时间计数器记录的当前时间的差值;

[0041] 所述提示单元,用于当存在差值大于预设值的目标局部时间计数器时,则对所述目标局部时间计数器所对应的工作进程进行报错提示。

[0042] 本发明实施例还提供了一种基于多线程任务的调度执行装置,包括:

[0043] 存储器,用于存储计算机程序;

[0044] 处理器,用于执行所述计算机程序以实现如上述任意一项所述基于多线程任务的调度执行方法的步骤。

[0045] 本发明实施例还提供了一种计算机可读存储介质,所述计算机可读存储介质上存储有计算机程序,所述计算机程序被处理器执行时实现如上述任意一项所述基于多线程任务的调度执行方法的步骤。

[0046] 由上述技术方案可以看出,将构建的定时器事件按照任务执行的时间顺序依次缓存至任务队列;其中,每个定时器事件有其对应的一个局部时间计数器以及工作线程;利用预先设定的工作线程更新全局时间计数器,并将全局时间计数器记录的当前时间作为所有定时器事件的局部时间计数器的初始时间。利用预先设定的工作线程维护全局时间计数器,可以有效的避免时区或系统时钟修改对全局时间计数器造成的影响。当第一个定时器事件的局部时间计数器记录的当前时间达到第一个定时器事件对应的任务执行时间时,则利用相应的工作线程执行第一个定时器事件所对应的待处理任务;将第一个定时器事件的局部时间计数器记录的当前时间作为下一个定时器事件的初始时间,直至轮询完任务队列中的所有定时器事件,则结束操作。下一个定时器事件依赖于与其相邻的上一个定时器事件的时间,保证了各定时器事件时间的相对同步,在不增加成本的前提下,解决了多线程任务调度执行过程中的时钟同步问题。并且在该技术方案中,将待处理任务以定时器事件的形式缓存至任务队列,针对每个定时器事件设置其对应的工作线程,通过定时器事件配合多线程工作队列进行触发监控以及任务的执行,提升了高并发I/O场景下的系统性能。

附图说明

[0047] 为了更清楚地说明本发明实施例,下面将对实施例中所需要使用的附图做简单的介绍,显而易见地,下面描述中的附图仅仅是本发明的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据这些附图获得其他的附图。

- [0048] 图1为本发明实施例提供的一种基于多线程任务的调度执行方法的流程图；
- [0049] 图2为本发明实施例提供的一种定时器事件链表的示意图；
- [0050] 图3为本发明实施例提供的一种基于多线程任务的调度执行装置的结构示意图；
- [0051] 图4为本发明实施例提供的一种基于多线程任务的调度执行装置的硬件结构示意图。

具体实施方式

[0052] 下面将结合本发明实施例中的附图，对本发明实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例仅仅是本发明一部分实施例，而不是全部实施例。基于本发明中的实施例，本领域普通技术人员在没有做出创造性劳动前提下，所获得的所有其他实施例，都属于本发明保护范围。

[0053] 为了使本技术领域的人员更好地理解本发明方案，下面结合附图和具体实施方式对本发明作进一步的详细说明。

[0054] 接下来，详细介绍本发明实施例所提供的一种基于多线程任务的调度执行方法。图1为本发明实施例提供的一种基于多线程任务的调度执行方法的流程图，该方法包括：

[0055] S101：将构建的定时器事件按照任务执行的时间顺序依次缓存至任务队列。

[0056] 其中，每个定时器事件有其对应的一个局部时间计数器以及工作线程。

[0057] 局部时间计数器用于记录定时器事件所对应的当前时间。

[0058] 在实际应用中，可以根据CPU核数创建相应数目的工作线程；为每个定时器事件分配相应的工作线程，以便于利用工作线程执行定时器事件所包含的待处理任务。

[0059] 在本发明实施例中，为了便于对所需执行的任务即待处理任务进行管理，可以基于待处理任务构建定时器事件；按照各定时器事件的触发时刻将所有定时器事件升序排列缓存至任务队列。

[0060] 触发时刻用于表征定时器事件所包含的待处理任务的执行时间。

[0061] 定时器事件中的待处理任务可能是一次性的任务，也可能是需要按照周期时间重复执行的任务，因此为了便于识别待处理任务是否需要重复执行，可以对定时器事件设置迭代周期，迭代周期用于表征定时器事件对应的待处理任务每隔多长时间执行一次。

[0062] 例如，定时器事件的迭代周期为0时，则表示对定时器事件包含的待处理任务执行一次即可。以时间精度为10ms为例，当迭代周期为5时，表示从第一次执行待处理任务开始，每隔50ms执行一次待处理任务。

[0063] 为了方便管理任务队列中的各定时器事件，定时器事件除了包含触发时刻和迭代周期外，还可以包括前后节点指向指针和增量触发时间。

[0064] 前后节点指向指针用于指示当前定时器事件与其相邻的前一个定时器事件以及与其相邻的后一个定时器事件的指向关系。

[0065] 对于每个定时器事件设置前后节点指向指针，可以便于定时器事件的管理，当有新的定时器事件需要放入任务队列时，依赖于前后节点指向指针便可以快速的确定新的定时器事件所需存放的位置。

[0066] 增量触发时间用于表征当前的定时器事件相对于其相邻的前一个定时器事件的触发时刻的差值。

[0067] S102:利用预先设定的工作线程更新全局时间计数器,并将全局时间计数器记录的当前时间作为所有定时器事件的局部时间计数器的初始时间。

[0068] 存储系统中包含有多个节点,每个节点有其对应的一个全局时间计数器。主控节点可以利用心跳机制,同步全系统所有节点的全局时间计数器,保证所有节点运行功能模块时钟一致,实现同步校时。

[0069] 在本发明实施例中,为了避免时区或系统时钟修改对全局时间计数器的时间造成影响,可以采用预先设定的工作线程对全局时间计数器进行更新维护。

[0070] 单一的工作线程可能由于高负载造成延时,导致全局时间计数器的更新不够及时,因此在实际应用中,可以采用多个高优先级的工作线程对全局时间计数器进行更新维护。

[0071] 为了避免多个高优先级的工作线程同时对全局时间计数器的时间进行更新,造成全局时间计数器更新的混乱,在具体实现中,可以按照锁保护的方式,利用多个高优先级工作线程更新全局时间计数器。

[0072] 通过设置锁保护方式,当多个高优先级工作线程中的任意一个工作线程对全局时间计数器进行时间的更新时,其它工作线程无法同时对该全局时间计数器执行更新操作。

[0073] 在本发明实施例中,全局时间计数器的更新周期可以根据实际需求设定,例如,可以每隔10ms对全局时间计数器的时间进行一次更新。

[0074] 初始状态下,可以将全局时间计数器记录的当前时间作为所有定时器事件的局部时间计数器的初始时间。

[0075] S103:当第一个定时器事件的局部时间计数器记录的当前时间达到第一个定时器事件对应的任务执行时间时,则利用相应的工作线程执行第一个定时器事件所对应的待处理任务。

[0076] 定时器事件按照任务执行的时间顺序在任务队列中排序,因此任务队列中的第一个定时器事件包含的任务为最早需要执行的任务。

[0077] 对于第一个定时器事件而言,由于其没有相邻的前一个定时器事件,因此,可以将第一个定时器事件的触发时刻作为任务执行时间。

[0078] 初始状态下,全局时间计数器完成时间的更新之后,会将其记录的当前时间作为第一个定时器事件的局部时间计数器的初始时间。此时,第一个定时器事件的局部时间计数器开始计时,当第一个定时器事件的局部时间计数器记录的当前时间达到第一个定时器事件的触发时刻时,则可以利用为第一定时器事件分配的工作线程执行第一个定时器事件所对应的待处理任务。

[0079] 在本发明实施例中,为了避免重复执行任务队列中记录的定时器事件,当完成第一个定时器事件所对应的待处理任务之后,可以将第一个定时器事件从任务队列中删除。

[0080] S104:将第一个定时器事件的局部时间计数器记录的当前时间作为下一个定时器事件的初始时间,直至轮询完任务队列中的所有定时器事件,则结束操作。

[0081] 在本发明实施例中,将各定时器事件按照任务执行的时间顺序排序放入任务队列中,因此对于每个定时器事件,可以依赖于其相邻的前一个定时器事件的时间执行任务。

[0082] 当第一个定时器事件的局部时间计数器记录的当前时间达到第一个定时器事件对应的任务执行时间时,此时可以将第一个定时器事件的局部时间计数器记录的当前时间

作为下一个定时器事件即第二个定时器事件的初始时间。相应的,第二个定时器事件对应的局部时间计数器开始计时,当第二个定时器事件的局部时间计数器记录的当前时间达到第二个定时器事件对应的任务执行时间时,则利用相应的工作线程执行第二个定时器事件所对应的待处理任务,依次类推,直至遍历完任务队列中的所有定时器事件则结束操作。

[0083] 参见图2所示的定时器事件链表,图2中左侧第一个框图用于表示全局时间计数器,剩余的每一个框图代表一个定时器事件,分别表示为Event1、Event2、Event3和Event4。全局时间计数器中的“count”表示当前已经对全局时间计数器时间进行更新的次数。在实际应用中,工作线程可以每隔10ms更新一次全局时间计数器记录的时间。每个定时器事件中的“time”表示增量触发时间。图2中是以4个定时器事件为例,第1个定时器事件发生在10个tick之后,1个tick表示10ms,也就是100ms后触发第1个定时器事件的待处理任务。第2个定时器事件在第1个定时器事件触发之后的5个tick之后触发,也就是 $100+5*10=150$ ms后触发第2个定时器事件的待处理任务。第3个定时器事件增量触发时间设置为0,说明它在第2个定时器事件触发之后立即触发。第4个定时器事件发生在第3个定时器事件触发之后的7个tick之后触发,也就是 $150+7*10=220$ ms后触发第1个定时器事件的待处理任务。

[0084] 由上述技术方案可以看出,将构建的定时器事件按照任务执行的时间顺序依次缓存至任务队列;其中,每个定时器事件有其对应的一个局部时间计数器以及工作线程;利用预先设定的工作线程更新全局时间计数器,并将全局时间计数器记录的当前时间作为所有定时器事件的局部时间计数器的初始时间。利用预先设定的工作线程维护全局时间计数器,可以有效的避免时区或系统时钟修改对全局时间计数器造成的影响。当第一个定时器事件的局部时间计数器记录的当前时间达到第一个定时器事件对应的任务执行时间时,则利用相应的工作线程执行第一个定时器事件所对应的待处理任务;将第一个定时器事件的局部时间计数器记录的当前时间作为下一个定时器事件的初始时间,直至轮询完任务队列中的所有定时器事件,则结束操作。下一个定时器事件依赖于与其相邻的上一个定时器事件的时间,保证了各定时器事件时间的相对同步,在不增加成本的前提下,解决了多线程任务调度执行过程中的时钟同步问题。并且在该技术方案中,将待处理任务以定时器事件的形式缓存至任务队列,针对每个定时器事件设置其对应的工作线程,通过定时器事件配合多线程工作队列进行触发监控以及任务的执行,提升了高并发IO场景下的系统性能。

[0085] 在实际应用中,待处理任务并非固定不变,随着任务队列中待处理任务的执行,与此同时也可能会存在新的待处理任务需要加入至任务队列中。

[0086] 在具体实现中,当接收到新的待处理任务时,对新的待处理任务构建新的定时器事件;根据新的定时器事件的触发时刻,将新的定时器事件插入任务队列。

[0087] 当在相邻的两个定时器事件之间插入新的定时器事件时,由于原本相邻的两个定时器事件的前后指向关系发生了变化,因此在将新的定时器事件插入任务队列之后,需要更新与新的定时器事件前后相邻的两个定时器事件的前后节点指向指针和增量触发时间。

[0088] 结合上述图2所示的定时器事件链表,假设新的定时器事件的触发时刻为180ms,此时需要将新的定时器事件插入原本第3个定时器事件以及第4个定时器事件中间,新插入的定时器事件的增量触发时间“time”记录为 $(180-150)/10=3$,原本的第4个定时器事件此时变为第5个定时器事件,其对应的增量触发时间“time”需要修改为 $(220-180)/10=4$ 。

[0089] 通过依赖于触发时刻,可以将新增加的定时器事件放入到任务队列的合适位置,

并且通过更新与新的定时器事件前后相邻的两个定时器事件的前后节点指向指针和增量触发时间,保证了任务队列中各定时器事件的参数信息的正确性,保证了各定时器事件的顺利执行。

[0090] 每个定时器事件的局部时间计数器通过工作线程维护,当某个定时器事件的工作线程出现问题时,会导致定时器事件的局部时间计数器更新的不及时,从而导致局部时间计数器记录的当前时间与全局时间计数器记录的当前时间出现较大的偏差。

[0091] 因此,在本发明实施例中,可以计算各局部时间计数器记录的当前时间与全局时间计数器记录的当前时间的差值;当存在差值大于预设值的目标局部时间计数器时,则说明与该目标局部时间计数器对应的工作线程可能已经阻塞,导致长时间未更新定时器事件的局部时间计数器,因此,当存在差值大于预设值的目标局部时间计数器时,可以对目标局部时间计数器所对应的工作进程进行报错提示。

[0092] 通过对局部时间计数器记录的时间与全局时间计数器记录的时间进行比较,可以及时发现局部时间计数器更新不及时的情况,从而对工作线程可能出现问题进行及时的报错提示,以便于管理人员及时处理。

[0093] 图3为本发明实施例提供的一种基于多线程任务的调度执行装置的结构示意图,包括缓存单元31、更新单元32、执行单元33和作为单元34;

[0094] 存储单元31,用于将构建的定时器事件按照任务执行的时间顺序依次缓存至任务队列;其中,每个定时器事件有其对应的一个局部时间计数器以及工作线程;

[0095] 更新单元32,用于利用预先设定的工作线程更新全局时间计数器,并将全局时间计数器记录的当前时间作为所有定时器事件的局部时间计数器的初始时间;

[0096] 执行单元33,用于当第一个定时器事件的局部时间计数器记录的当前时间达到第一个定时器事件对应的任务执行时间时,则利用相应的工作线程执行第一个定时器事件所对应的待处理任务;

[0097] 作为单元34,用于将第一个定时器事件的局部时间计数器记录的当前时间作为下一个定时器事件的初始时间,直至轮询完任务队列中的所有定时器事件,则结束操作。

[0098] 可选地,缓存至缓存单元包括构建子单元和排列子单元;

[0099] 构建子单元,用于基于待处理任务构建定时器事件;其中,每个定时器事件包括前后节点指向指针、触发时刻、迭代周期和增量触发时间;

[0100] 排列子单元,用于按照各定时器事件的触发时刻将所有定时器事件升序排列缓存至任务队列。

[0101] 可选地,还包括构建单元、插入单元和指针更新单元;

[0102] 构建单元,用于当接收到新的待处理任务时,对新的待处理任务构建新的定时器事件;

[0103] 插入单元,用于根据新的定时器事件的触发时刻,将新的定时器事件插入任务队列;

[0104] 指针更新单元,用于更新与新的定时器事件前后相邻的两个定时器事件的前后节点指向指针和增量触发时间。

[0105] 可选地,更新单元包括按照锁保护的方式,利用多个高优先级工作线程更新全局时间计数器。

- [0106] 可选地,还包括删除单元;
- [0107] 删除单元,用于当完成第一个定时器事件所对应的待处理任务之后,将第一个定时器事件从任务队列中删除。
- [0108] 可选地,还包括计算单元和提示单元;
- [0109] 计算单元,用于计算各局部时间计数器记录的当前时间与全局时间计数器记录的当前时间的差值;
- [0110] 提示单元,用于当存在差值大于预设值的目标局部时间计数器时,则对目标局部时间计数器所对应的工作进程进行报错提示。
- [0111] 图3所对应实施例中特征的说明可以参见图1所对应实施例的相关说明,这里不再一一赘述。
- [0112] 由上述技术方案可以看出,将构建的定时器事件按照任务执行的时间顺序依次缓存至任务队列;其中,每个定时器事件有其对应的一个局部时间计数器以及工作线程;利用预先设定的工作线程更新全局时间计数器,并将全局时间计数器记录的当前时间作为所有定时器事件的局部时间计数器的初始时间。利用预先设定的工作线程维护全局时间计数器,可以有效的避免时区或系统时钟修改对全局时间计数器造成的影响。当第一个定时器事件的局部时间计数器记录的当前时间达到第一个定时器事件对应的任务执行时间时,则利用相应的工作线程执行第一个定时器事件所对应的待处理任务;将第一个定时器事件的局部时间计数器记录的当前时间作为下一个定时器事件的初始时间,直至轮询完任务队列中的所有定时器事件,则结束操作。下一个定时器事件依赖于与其相邻的上一个定时器事件的时间,保证了各定时器事件时间的相对同步,在不增加成本的前提下,解决了多线程任务调度执行过程中的时钟同步问题。并且在该技术方案中,将待处理任务以定时器事件的形式缓存至任务队列,针对每个定时器事件设置其对应的工作线程,通过定时器事件配合多线程工作队列进行触发监控以及任务的执行,提升了高并发I0场景下的系统性能。
- [0113] 图4为本发明实施例提供的一种基于多线程任务的调度执行装置40的硬件结构示意图,包括:
- [0114] 存储器41•,用于存储计算机程序;
- [0115] 处理器42,用于执行计算机程序以实现如上述任意实施例所述的基于多线程任务的调度执行方法的步骤。
- [0116] 本发明实施例还提供了一种计算机可读存储介质,计算机可读存储介质上存储有计算机程序,计算机程序被处理器执行时实现如上述任意实施例所述的基于多线程任务的调度执行方法的步骤。
- [0117] 以上对本发明实施例所提供的一种基于多线程任务的调度执行方法、装置和计算机可读存储介质进行了详细介绍。说明书中各个实施例采用递进的方式描述,每个实施例重点说明的都是与其他实施例的不同之处,各个实施例之间相同相似部分互相参见即可。对于实施例公开的装置而言,由于其与实施例公开的方法相对应,所以描述的比较简单,相关之处参见方法部分说明即可。应当指出,对于本技术领域的普通技术人员来说,在不脱离本发明原理的前提下,还可以对本发明进行若干改进和修饰,这些改进和修饰也落入本发明权利要求的保护范围内。
- [0118] 专业人员还可以进一步意识到,结合本文中所公开的实施例描述的各示例的单元

及算法步骤,能够以电子硬件、计算机软件或者二者的结合来实现,为了清楚地说明硬件和软件的可互换性,在上述说明中已经按照功能一般性地描述了各示例的组成及步骤。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发明的范围。

[0119] 结合本文中所公开的实施例描述的方法或算法的步骤可以直接用硬件、处理器执行的软件模块,或者二者的结合来实施。软件模块可以置于随机存储器(RAM)、内存、只读存储器(ROM)、电可编程ROM、电可擦除可编程ROM、寄存器、硬盘、可移动磁盘、CD-ROM、或技术领域内所公知的任意其它形式的存储介质中。

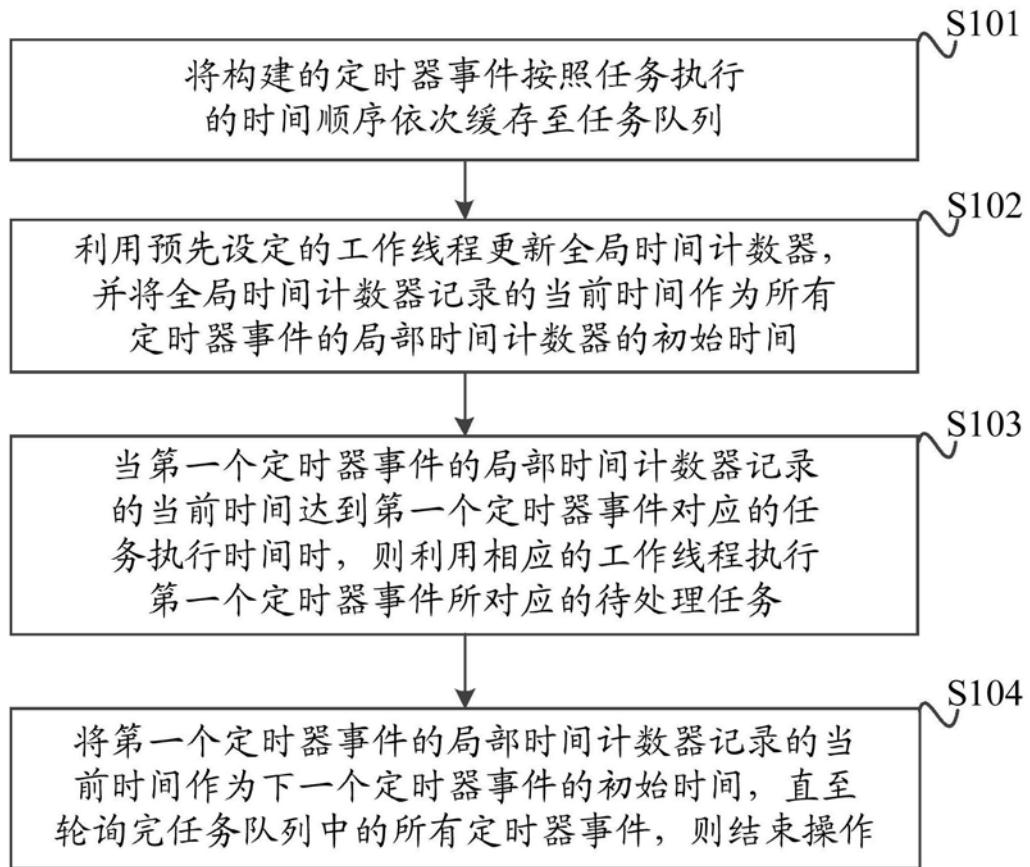


图1

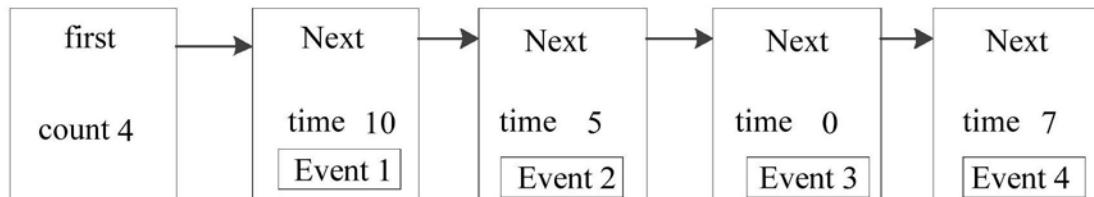


图2

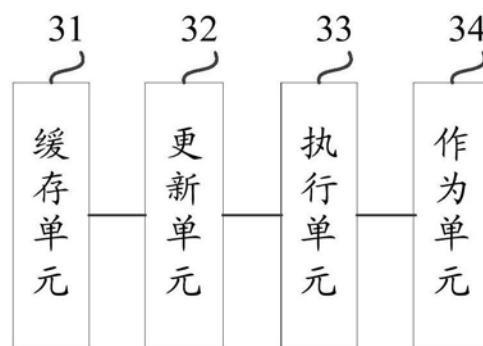


图3

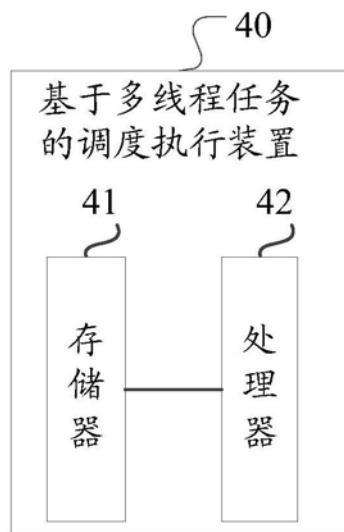


图4