



(12) 发明专利

(10) 授权公告号 CN 102629221 B

(45) 授权公告日 2014. 11. 19

(21) 申请号 201210047777. 4

CN 1804836 A, 2006. 07. 19, 全文.

(22) 申请日 2012. 02. 28

CN 101631328 A, 2010. 01. 20, 全文.

(73) 专利权人 华为技术有限公司

CN 1268688 A, 2200. 10. 04, 全文.

地址 518129 广东省深圳市龙岗区坂田华为  
总部办公楼

审查员 赵晓敏

(72) 发明人 顾磷 马志强 曾毓珑

(74) 专利代理机构 北京中博世达专利商标代理  
有限公司 11274

代理人 申健

(51) Int. Cl.

G06F 9/52 (2006. 01)

(56) 对比文件

CN 102346740 A, 2012. 02. 08, 说明书第  
[0005]-[0067] 段、图 1-3.

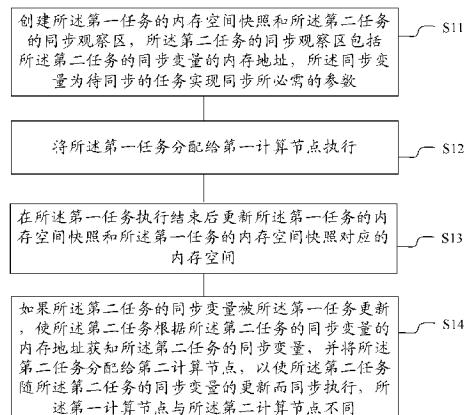
权利要求书2页 说明书10页 附图3页

(54) 发明名称

用于分布式共享存储的任务同步方法、装置  
及系统

(57) 摘要

本发明的实施例提供了用于分布式共享存储的任务同步方法、装置及系统，涉及分布式共享存储领域，为简化程序设计同时提升处理性能而发明。所述方法包括：创建第一任务的内存空间快照和第二任务的同步观察区，所述第二任务的同步观察区包括所述第二任务的同步变量的内存地址，所述同步变量为待同步的任务实现同步所必需的参数；将所述第一任务分配给第一计算节点执行；在所述第一任务执行结束后更新所述第一任务的内存空间快照及其对应的内存空间；如果所述第二任务的同步变量被所述第一任务更新，使所述第二任务根据所述第二任务的同步变量的内存地址获知所述第二任务的同步变量，并将所述第二任务分配给第二计算节点。



1. 一种用于分布式共享存储的任务同步方法，其特征在于，包括：

创建第一任务的内存空间快照和第二任务的同步观察区，所述第二任务的同步观察区包括所述第二任务的同步变量的内存地址，所述同步变量为待同步的任务实现同步所必需的参数，与所述第一任务相关的内存空间为共享空间；

将所述第一任务分配给第一计算节点执行；

在所述第一任务执行结束后，更新所述第一任务的内存空间快照和所述第一任务的内存空间快照对应的内存空间；

如果所述第二任务的同步变量被所述第一任务更新，使所述第二任务根据所述第二任务的同步观察区中的同步变量的内存地址获知所述第二任务的同步变量，并将所述第二任务分配给第二计算节点，以使所述第二任务随所述第二任务的同步变量的更新而同步执行；

在所述创建所述第一任务的内存空间快照和所述第二任务的同步观察区后，所述方法还包括：

将所述第二任务放入观察者队列，所述观察者队列为存放具有同步观察区、但不能被立即执行的任务的队列；

所述将所述第二任务分配给第二计算节点包括：

将所述第二任务从所述观察者队列移至执行队列后，将所述第二任务分配给所述第二计算节点，所述执行队列为存放即将被执行的任务的队列；

所述将所述第二任务从所述观察者队列移至执行队列包括：

将所述观察者队列中包括所述第二任务在内的、所有同步变量被所述第一任务更新的任务，从所述观察者队列移至所述执行队列；

所述将所述第二任务分配给第二计算节点，以使所述第二任务随所述第二任务的同步变量的更新而同步执行包括：

如果所述第二任务的同步变量中还存在至少一个同步变量所对应的内存空间没有被更新，则将所述第二任务重新放入所述观察者队列；

如果所述第二任务的同步变量所对应的内存空间均已被更新，则执行所述第二任务。

2. 根据权利要求 1 所述的方法，其特征在于，所述更新所述第一任务的内存空间快照和所述第一任务的内存空间快照对应的内存空间包括：

如果所述第一任务的内存空间快照对应的内存空间已经被所述第一任务以外的任务更新，则放弃对所述第一任务的内存空间快照及其对应的内存空间的更新；

如果所述第一任务的内存空间快照对应的内存空间尚未被所述第一任务以外的任务更新，则更新所述第一任务的内存空间快照及其对应的内存空间。

3. 根据权利要求 1 所述的方法，其特征在于，所述第二任务的同步观察区位于所述第二任务的运行上下文中。

4. 一种用于分布式共享存储的任务同步装置，其特征在于，包括：

创建单元，用于创建第一任务的内存空间快照和第二任务的同步观察区，所述第二任务的同步观察区包括所述第二任务的同步变量的内存地址，所述同步变量为待同步的任务实现同步所必需的参数，与所述第一任务相关的内存空间为共享空间；

更新单元，用于将所述第一任务分配给第一计算节点执行，在所述第一任务执行结束

后更新所述第一任务的内存空间快照和所述第一任务的内存空间快照对应的内存空间；

同步单元，用于当所述第二任务的同步变量被所述第一任务更新时，使所述第二任务根据所述第二任务的同步观察区中的同步变量的内存地址获知所述第二任务的同步变量，并将所述第二任务分配给第二计算节点，以使所述第二任务随所述第二任务的同步变量的更新而同步执行；

观察者调度单元，用于将所述第二任务放入观察者队列，所述观察者队列为存放具有同步观察区、但不能被立即执行的任务的队列；

所述同步单元包括：

第一同步子单元，用于当所述第二任务的同步变量被所述第一任务更新时，使所述第二任务根据所述第二任务的同步观察区中的同步变量的内存地址获知所述第二任务的同步变量；

第二同步子单元，用于在将所述第二任务从所述观察者队列移至执行队列后，将所述第二任务分配给所述第二计算节点，所述执行队列为存放即将被执行的任务的队列；

所述第二同步子单元具体用于：

在将所述观察者队列中包括所述第二任务在内的、所有同步变量被所述第一任务更新的任务，从所述观察者队列移至所述执行队列后，将所述第二任务分配给所述第二计算节点；

所述第二同步子单元具体用于：

在将所述第二任务从所述观察者队列移至执行队列后，如果所述第二任务的同步变量中还存在至少一个同步变量所对应的内存空间没有被更新，则将所述第二任务重新放入所述观察者队列；

在将所述第二任务从所述观察者队列移至执行队列后，如果所述第二任务的同步变量所对应的内存空间均已被更新，则执行所述第二任务。

5. 根据权利要求 4 所述的装置，其特征在于，所述更新单元包括：

第一更新子单元，用于当所述第一任务的内存空间快照对应的内存空间已经被所述第一任务以外的任务更新时，放弃对所述第一任务的内存空间快照及其对应的内存空间的更新；

第二更新子单元，用于当所述第一任务的内存空间快照对应的内存空间尚未被所述第一任务以外的任务更新时，更新所述第一任务的内存空间快照及其对应的内存空间。

6. 一种分布式共享存储系统，其特征在于，包括第一计算节点、第二计算节点以及权利要求 4-5 中任一项所述的用于分布式共享存储的任务同步装置；所述用于分布式共享存储的任务同步装置位于所述第一计算节点内，或者位于所述第二计算节点内，或者分别与所述第一计算节点和所述第二计算节点相连，用于将所述第一计算节点与所述第二计算节点中的需要同步的任务进行同步。

## 用于分布式共享存储的任务同步方法、装置及系统

### 技术领域

[0001] 本发明涉及分布式共享存储领域，尤其涉及一种用于分布式共享存储的任务同步方法、装置及系统。

### 背景技术

[0002] 当前，并行计算机的存储结构可以大致分为：共享存储结构、分布式存储结构。在共享存储结构中，所有处理器都有一致访问的全局物理内存，支持全局共享变量的编程模型。其编程简单，但受到共享内存带宽等的限制，扩展性较差。在分布式存储结构中，许多独立的有本地存储的计算节点通过高速网络互联，每个计算节点有单独的地址空间。计算节点间通过显式的消息传递使各个计算节点所执行的任务之间进行通信。其中，任务是系统进行资源分配和调度执行的基本单位，包含数据和对数据的操作序列。多个任务可以相互配合、并发执行，从而共同实现特定功能。分布式存储结构的扩展性能好，但由于需考虑数据分配和消息的传递，其程序设计较困难。

[0003] DSM(Distributed Shared Memory, 分布式共享存储)结构，在物理存储分散的系统上通过硬件或软件实现了逻辑上的共享存储。在 DSM 中特别是通过软件实现的 DSM 中，底层的消息传递机制对用户掩盖起来，允许用户以共享存储方式进行并行程序设计。由于分布式共享存储系统既具有共享存储系统易于编程的优点，又保留了分布式存储系统的可扩展性，因而是大规模并行计算系统的一种重要形式。在 DSM 系统中，当存在多个任务（例如多个进程或者线程）共同实现特定的功能，且各任务之间存在前后制约关系、需要遵守某种顺序约束时，各任务的执行需要同步。因此，如何实现任务的同步，是并行程序设计必须要解决的关键问题。

[0004] 现有技术中的锁、信号量、管程等同步手段仅适用于具有公共存储区的单机环境。现有技术中的基于消息传递编程模式的路障同步法，即在参与路障同步的每个任务的程序中彼此必须等待的位置设置一个障碍点，当某任务执行到障碍点时暂停，等待所有任务都执行到这个障碍点后，该任务才能继续运行。显式的消息传递编程要求程序员关心数据的划分和任务间的通信，因此在解决数据依赖和预防死锁方面花费大量力气，容易出错。

### 发明内容

[0005] 本发明实施例提供一种用于分布式共享存储的任务同步方法、装置及系统，能够简化程序设计同时提升系统的处理性能。本发明提供了如下技术方案：

[0006] 一方面，本发明实施例提供一种用于分布式共享存储的任务同步方法，包括：

[0007] 创建所述第一任务的内存空间快照和所述第二任务的同步观察区，所述第二任务的同步观察区包括所述第二任务的同步变量的内存地址，所述同步变量为待同步的任务实现同步所必需的参数；

[0008] 将所述第一任务分配给第一计算节点执行；

[0009] 在所述第一任务执行结束后，更新所述第一任务的内存空间快照和所述第一任务

的内存空间快照对应的内存空间；

[0010] 如果所述第二任务的同步变量被所述第一任务更新，使所述第二任务根据所述第二任务的同步观察区中的同步变量的内存地址获知所述第二任务的同步变量，并将所述第二任务分配给第二计算节点，以使所述第二任务随所述第二任务的同步变量的更新而同步执行。

[0011] 另一方面，本发明实施例提供一种用于分布式共享存储的任务同步装置，包括：

[0012] 创建单元，用于创建第一任务的内存空间快照和第二任务的同步观察区，所述第二任务的同步观察区包括所述第二任务的同步变量的内存地址，所述同步变量为待同步的任务实现同步所必需的参数；

[0013] 更新单元，用于将所述第一任务分配给第一计算节点执行，并在所述第一任务执行结束后更新所述第一任务的内存空间快照和所述第一任务的内存空间快照对应的内存空间；

[0014] 同步单元，用于当所述第二任务的同步变量被所述第一任务更新时，使所述第二任务根据所述第二任务的同步观察区中的同步变量的内存地址获知所述第二任务的同步变量，并将所述第二任务分配给第二计算节点，以使所述第二任务随所述第二任务的同步变量的更新而同步执行。

[0015] 另一方面，本发明实施例提供一种分布式共享存储系统，包括第一计算节点、第二计算节点以及本发明实施例提供的用于分布式共享存储的任务同步装置；所述用于分布式共享存储的任务同步装置位于所述第一计算节点内，或者位于所述第二计算节点内，或者分别与所述第一计算节点和所述第二计算节点相连，用于将所述第一计算节点与所述第二计算节点中的需要同步的任务进行同步。

[0016] 采用上述技术方案后，本发明实施例提供的用于分布式共享存储的任务同步方法、装置及系统，创建了第一任务的内存空间快照和第二任务的同步观察区，所述第二任务的同步观察区存储有所述第二任务的同步变量的内存地址。一方面，第一任务在计算节点本地执行，任务完成后才更新共享存储空间，另一方面，如果所述第二任务的同步变量被所述第一任务更新，可以使所述第二任务根据所述第二任务的同步变量的内存地址获知所述第二任务的同步变量，并将所述第二任务分配给第二计算节点，以使所述第二任务随所述第二任务的同步变量的更新而同步执行，在很好地维护了分布式共享存储系统的内存一致性即程序对内存操作的正确性的同时，大大减少了分布式共享存储的任务同步和内存一致性维护所需要的频繁的消息发送和状态探寻，既简化了编程又有效提高了系统的处理性能。

## 附图说明

[0017] 为了更清楚地说明本发明实施例或现有技术中的技术方案，下面将对实施例描述中所需要使用的附图作简单地介绍，显而易见地，下面描述中的附图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。

[0018] 图 1 为本发明实施例提供的用于分布式共享存储的任务同步方法的一种流程图；

[0019] 图 2 为本发明实施例提供的用于分布式共享存储的任务同步方法的一种详细流

程图；

[0020] 图3为本发明实施例提供的用于分布式共享存储的任务同步装置的一种结构示意图；

[0021] 图4为本发明实施例提供的用于分布式共享存储的任务同步装置中的更新单元的结构示意图；

[0022] 图5为本发明实施例提供的用于分布式共享存储的任务同步装置的另一种结构示意图。

### 具体实施方式

[0023] 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，所描述的实施例是本发明一部分实施例，而不是全部的实施例。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，都属于本发明保护的范围。

[0024] 如图1所示，本发明的实施例提供了一种用于分布式共享存储的任务同步方法，包括：

[0025] S11，创建第一任务的内存空间快照和第二任务的同步观察区，所述第二任务的同步观察区包括所述第二任务的同步变量的内存地址，所述同步变量为待同步的任务实现同步所必需的参数；

[0026] S12，将所述第一任务分配给第一计算节点执行；

[0027] S13，在所述第一任务执行结束后，更新所述第一任务的内存空间快照和所述第一任务的内存空间快照对应的内存空间；

[0028] S14，如果所述第二任务的同步变量被所述第一任务更新，使所述第二任务根据所述第二任务的同步观察区中的同步变量的内存地址获知所述第二任务的同步变量，并将所述第二任务分配给第二计算节点，以使所述第二任务随所述第二任务的同步变量的更新而同步执行。

[0029] 采用上述技术方案后，本发明实施例提供的用于分布式共享存储的任务同步方法，创建了第一任务的内存空间快照和第二任务的同步观察区，所述第二任务的同步观察区存储有所述第二任务的同步变量的内存地址。一方面，第一任务在计算节点本地执行，任务完成后才更新共享存储空间，另一方面，如果所述第二任务的同步变量被所述第一任务更新，可以使所述第二任务根据所述第二任务的同步变量的内存地址获知所述第二任务的同步变量，并将所述第二任务分配给第二计算节点，以使所述第二任务随所述第二任务的同步变量的更新而同步执行，在很好地维护了分布式共享存储系统的内存一致性即程序对内存操作的正确性的同时，大大减少了分布式共享存储的任务同步和内存一致性维护所需要的频繁的消息发送和状态探寻，既简化了编程又有效提高了系统的处理性能。

[0030] 需要说明的是，在分布式共享存储结构中，存在多个计算节点。本实施例中的第一计算节点和第二计算节点可以相同，也可以不同，本发明对此不做限制。不同的任务，既可以分配给不同计算节点执行，也可以分配给同一个节点的不同处理器执行。具体的，在本发明的一个实施例中，当第一计算节点上的任务执行完成后，分配下一个任务时，需要选择一个空闲的计算节点执行该任务，此时既可能选择其他的计算节点，也可能选择当前空闲的

第一计算节点。

[0031] 当某个任务的执行需要另一个或另外多个任务的执行结果时,就需要知道所需要的另外一个或另外多个任务是否执行完毕,它们的执行结果是否可以被需要同步的任务使用了,即本发明所述的任务同步问题。

[0032] 具体的,本实施例中的任务可以包含控制信息以及与任务相关的数据和代码。其中,控制信息可以包括任务执行相关的堆栈信息、同步观察区和首条指令地址等。分配在不同计算节点的任务从首条指令开始执行,在执行过程中发现数据不在本地则从全局共享内存中获取,任务成功执行后有可能修改全局内存以实现数据的更新或后续任务的同步。

[0033] 为了便于说明,本实施例中,假定第一任务可以不受其它条件的制约而执行,或者说所述第一任务的执行条件已经具备。而第二任务是需要依赖于其它任务的执行结果才能执行。

[0034] 还需要说明的是,本发明虽然以两个任务的任务同步方法为例进行说明,但本发明不限于此,所述任务同步可以是三个或三个以上任务的同步,其原理与两个任务的任务同步方法类似。

[0035] 具体的,在步骤 S11 中,创建第一任务的内存空间快照和第二任务的同步观察区。可选的,第一任务的内存空间快照可以为基于事务内存的内存空间快照。此处,与第一任务相关的内存空间为共享空间。

[0036] 所述第二任务的同步观察区包含所述第二任务的同步变量的内存地址。其中,所述同步变量为待同步的任务实现同步所必需的参数。可选的,所述第二任务的同步观察区可以位于所述第二任务的运行上下文中。

[0037] 举例说明,在本发明的一个实施例中,在执行第二任务之前,需要初始化第二任务的运行上下文,这时,可以在第二任务的运行上下文中增加一片区域作为第二任务的同步观察区。在第二任务的同步观察区中,可以存放第二任务的同步变量的地址。只有通过该地址获得同步变量后,第二任务才能执行。

[0038] 具体的,在步骤 S12 中,第一任务的执行条件已经具备,因此将第一任务分配给第一计算节点执行。需要说明的是,在执行第一任务的过程中,执行第一任务的计算节点只对第一任务的内存空间快照的一个副本进行写操作,而不会更新内存空间快照及其对应的内存空间。此处,第一任务的内存空间快照的一个副本为计算节点的本地内存。也就是说,第一任务的执行过程中,只更新计算节点的本地内存而不更新共享内存。

[0039] 而在第一任务执行结束后在 S13 步骤中,在第一任务执行结束后,更新所述第一任务的内存空间快照和所述第一任务的内存空间快照对应的内存空间。

[0040] 在 S13 步骤后,本实施例提供的用于分布式共享存储的任务同步方法可以根据各个需要同步的任务的同步变量,确定任意任务的执行结果,如第一任务的执行结果,是否为另外一个任务的同步变量。如果一任务的执行结果是另一个任务的同步变量,则每当所述一任务执行完成,更新了该任务的内存空间快照及其对应的内存空间后,由于该任务的执行结果,即所述另一个任务的同步变量,也存储在被更新了的内存空间中,因此,所述另一个任务的同步变量也就被更新了,从而使所述另一个任务能够根据其同步观察区中同步变量的地址获知更新后的同步变量。而如果某一任务没有执行完毕,就无法使其它任务获知与所述某一任务相关的同步变量。

[0041] 具体的,在步骤 S14 中,如果所述第二任务的同步变量被所述第一任务更新,使所述第二任务根据所述第二任务的同步变量的内存地址获知所述第二任务的同步变量,并将所述第二任务分配给第二计算节点,以使所述第二任务随所述第二任务的同步变量的更新而同步执行。

[0042] 例如,在本发明的一个实施例中,第二任务的执行需要同步变量 S,在所述第二任务的同步观察区中存储有参数 S 在内存中的地址 addr(S)。S 同时又是第一任务的执行结果,如果第一任务尚未执行结束,其执行结果 S 就是未知的,addr(S) 中的 S 也没有被更新,无法使第二任务获知其同步变量,在这种情况下,第二任务自然无法执行。

[0043] 而当第一任务执行结束后,就可以更新所述第一任务的内存空间快照和所述第一任务的内存空间快照对应的内存空间。这样,存储在内存地址 addr(S) 中的参数 S 即被更新,从而使第二任务通过其同步观察区中存储的 addr(S) 获知所述第二任务的同步变量 S,进而使第二任务得以伴随 S 的更新而同步执行。

[0044] 具体的,对于内存空间快照的更新操作具有原子性。也就是说,对内存快照的更新,或者全部都更新,或者全部都不更新。而在更新所述第一任务的内存空间快照对应的内存空间时,又可能遇到两种情况。

[0045] 其一,如果在更新所述第一任务的内存空间快照对应的内存空间时,所述第一任务的内存空间快照对应的内存空间已经被所述第一任务以外的任务更新,则放弃对所述内存空间快照及其对应的内存空间的更新。其二,如果在更新所述第一任务的内存空间快照对应的内存空间时,所述第一任务的内存空间快照对应的内存空间尚未被所述第一任务以外的任务更新,则更新所述内存空间快照及其对应的内存空间。

[0046] 上述情况的发生是因为第一任务的内存空间快照对应的内存空间为共享存储空间,该内存空间所存储的内容的更新不仅可以由第一任务进行,也可以由第一任务和第二任务以外的其它任务进行。因此,如果在第一任务执行结束后,需要更新第一任务的内存空间快照及其对应的内存空间,则首先要确定,第一任务的内存空间快照对应的内存空间是否已经被其它任务更新过。具体的确定方法可以有多种,本发明对此不做限定。例如,在本发明的一个实施例中,可以通过确定此时的内存空间的存储状态(如版本号信息)是否与第一任务执行前的内存空间的存储状况(如第一任务的内存空间快照的状况)相同的方法进行确定。

[0047] 具体的,在本发明的一个实施例中,第一任务的内存空间快照对应的内存空间已经被所述第一任务以外的任务更新,则放弃对所述第一任务的内存空间快照及其对应的内存空间的更新。此时,优选的,可以重新创建第一任务,以使第一任务可以被重新执行,从而实现对第一任务的内存空间快照可能的更新。如果所述第一任务的内存空间快照对应的内存空间尚未被所述第一任务以外的任务更新,则更新所述内存空间快照及其对应的内存空间。

[0048] 进一步地,在本发明的另一个实施例中,第二任务的同步参数没有被所述第二任务更新,也就是说,本实施例中,第一任务的执行结果并没有对其它任务(本实施例中为第二任务)的执行产生约束,其它任务不需要获知第一任务的执行结果即可执行,则可以在所述更新所述第一任务的内存空间快照和所述第一任务的内存空间快照对应的内存空间后删除所述第一任务的内存空间快照。这样,由于不涉及任务间的同步问题,及时将内存空

间快照删除后可以释放更多的内存空间。

[0049] 需要说明的是,前述实施例中,第一任务执行完毕后会更新所述第一任务的内存空间快照及其对应的内存空间,如果第二任务的同步参数也存储在所述第一任务的内存空间快照所对应的内存空间,则,所述第二任务的同步参数也被更新,所述第二任务即被激活。但本发明不限于此,第一任务的执行结果可能涉及不止一个任务的同步执行。

[0050] 进一步地,在本发明的另一个实施例中,如果分布式共享存储结构中存在多个需要同步的任务,即存在多个具有同步观察区的任务,则可以为这些任务的调度设置一个观察者队列,将所述第二任务放入所述观察者队列。所述观察者队列为存放具有同步观察区、但不能被立即执行的任务的队列。

[0051] 此时,所述第一任务的内存空间快照对应的内存空间的更新,能够使所述观察者队列中的第二任务通过同步观察区中同步变量的地址获知其同步变量被更新。于是,所述第二任务即被激活,并从观察者队列移至执行队列,等待被调度执行。所述执行队列为存放即将被执行的任务的队列。

[0052] 具体的,本实施例中,第一任务执行完毕并更新所述第一任务的内存空间快照及其对应的内存空间后,可以通过检索所述观察队列中的每一个任务的同步观察区的方法,获知所述第二任务的同步参数以及所述观察队列中的其它任务的同步参数是否被所述第一任务更新,并将所述观察者队列中包括所述第二任务在内的、所有同步变量被所述第一任务更新的任务,从所述观察者队列移至所述执行队列。以此保证所有需要与第一任务的执行结果相同步的任务都能通过该方法同步。

[0053] 举例说明,在本发明的一个实施例中,任务 B、任务 C 和任务 D 均在观察者队列中,其中,任务 B 和任务 C 的执行依赖任务 A 的执行结果,即任务 B 和任务 C 的同步变量包括任务 A 的执行结果,而任务 D 的执行不依赖任务 A 的执行结果。则任务 A 执行完毕后,将任务 A 的内存空间快照及其对应的内存空间进行更新,该更新使任务 B 和任务 C 获知其同步变量已经被更新,从而使任务 B 和任务 C 由观察者队列移至执行队列等待执行。而由于任务 D 的同步变量没有被更新,任务 D 仍然保留在观察者队列中。

[0054] 将所述第二任务放置于观察者队列后,本实施例中,所述将所述第二任务分配给第二计算节点,以使所述第二任务随所述第二任务的同步变量的更新而同步执行,具体可以为:如果所述第二任务的同步变量中还存在至少一个同步变量所对应的内存空间没有被更新,则将所述第二任务重新放入所述观察者队列;如果所述第二任务的同步变量所对应的内存空间均已被更新,则执行所述第二任务。

[0055] 举例说明,在本发明的一个实施例中,第二任务的执行需要第一任务的执行结果 R1 以及第三任务的执行结果 R3,即第二任务的同步变量包括 R1 和 R3。第二任务被创建后,首先被放入观察者队列。如果第一任务首先执行完毕,更新了其内存空间快照及其对应的内存空间,相应的,R1 被更新,则可使第二任务通过其同步观察区中的同步变量的内存地址获知更新后的同步变量 R1,第二任务即被激活,并由观察者队列移至执行队列,等待被调度执行。所述第二任务被调度出执行队列后即被执行,此时如果发现第二任务的同步变量中,所述第三任务的执行结果 R3 并未被更新,则停止执行第二任务,并将第二任务重新放入观察者队列。当第三任务执行完毕,其执行结果 R3 被更新到第三任务的内存空间快照及其对应的内存空间后,可使第二任务获知其同步变量 R3 被更新,第二任务将再次被激活并从观

察者队列移至执行队列等待执行。此时,由于其同步变量 R1 和 R3 都已经更新,当第二任务被调度出执行队列时,所述第二任务即可以被执行。

[0056] 本发明实施例的方法可以由通用集成电路,如 CPU(Central Processing Unit, 中央处理器)或 ASIC(Application Specific Integrated Circuit, 专用集成电路)等执行。

[0057] 为了使本领域的技术人员更好的理解本发明的技术方案,下面通过具体的实施例对本发明实施例的具体技术方案进行详细描述,可以理解的是,以下的具体实施例仅用于描述本发明,但本发明不限于此。

[0058] 图 2 所示为本发明提供的用于分布式共享存储的任务同步方法的一个具体实施例的方法流程图。如图 2 所示,所述方法具体可包括:

[0059] 101. 创建任务,设置所述任务的运行上下文;

[0060] 102. 确定是否需要创建同步观察区,如果是,则执行步骤 103,如果否,则执行步骤 104;

[0061] 具体的,可以确定该任务的执行是否需要与其它任务同步,如果需要,则该任务需要创建同步观察区,以便获知其同步变量是否被更新;如果不需,则该任务不需要创建同步观察区。

[0062] 103. 创建观察者任务,将该观察者任务放入观察者队列;

[0063] 其中,所述观察者任务为需要同步变量的同步才能执行的任务。

[0064] 1031. 观察者任务被激活成为普通任务(以下简称任务),将该任务放入执行队列;执行步骤 105;

[0065] 104. 将该任务放入执行队列,等待调度;

[0066] 105. 分配任务到计算节点执行;

[0067] 106. 任务执行结束后,确定该任务的内存空间是否已被其它任务更新;如果是,放弃该任务的本次所有操作并重启该任务,执行步骤 105;如果否,则执行步骤 107;

[0068] 107. 更新该任务的内存空间快照及其对应的内存空间;

[0069] 108. 确定观察者队列中是否存在同步变量被更新的观察者任务;如果是,则执行步骤 1031;如果否,则执行步骤 109;

[0070] 109. 确定执行队列是否为空;如果是,则执行步骤 110,如果否,则跳转至步骤 105;

[0071] 110. 显示执行结果;

[0072] 111. 结束。

[0073] 需要说明的是,本发明实施例提供的用于分布式共享存储的任务同步方法,可以使一个任务通过其同步观察区获知所述任务的同步变量是否已经更新并根据更新后的同步变量执行所述任务,在很好地维护了分布式共享存储系统的内存一致性即程序对内存操作的正确性的同时,大大减少了分布式共享存储的任务同步和内存一致性维护所需要的频繁的消息发送和状态探寻,既简化了编程又有效提高了系统的处理性能。

[0074] 又例如,本发明实施例提供的用于分布式共享存储的任务同步方法,能有效地处理生产者-消费者问题。简单起见,假设缓冲区满缓冲单元的数目为 n(初始为 0),生产者生产一个产品对应于 n 值的加 1,消费者消费一个产品对应于 n 值的减 1。使用变量 m 控制生产者、消费者对 n 的更新:m 为 1 时,允许消费者访问,生产者等待;m 为 0 时,允许生产者

访问,消费者等待。

[0075] 假定变量 m 的初始值为 0,并且生产者任务和生产者的观察者任务同时创建,其具体的伪代码如表 1 所示:

[0076] 表 1

[0077]

<pre> producer_runner:     if (m == 1)         exit and abort     else         n = n+1         m = 1         exit and commit     </pre>	<pre> consumer_runner:     if (m == 0)         exit and abort     else         n = n-1         m = 0         exit and commit     </pre>
<pre> producer_watcher:     watches m and n     if (m == 1) then         exit and abort     else         create producer_runner         create producer_watcher         exit and commit     </pre>	<pre> consumer_watcher:     watches m and n     if (m == 0) then         exit and abort     else         create consumer_runner         create consumer_watcher         exit and commit     </pre>

[0078] 具体的, producer\_runner 为生产者任务, producer\_watcher 为生产者的观察者任务, consumer\_runner 为消费者任务, consumer\_watcher 为消费者的观察者任务。其中, producer\_runner 和 consumer\_runner 为普通任务;而 producer\_watcher 和 consumer\_watcher 是具有同步观察区的观察者任务, 创建后不能被直接放入执行队列中执行而要放入观察者队列中,通过其同步观察区中的同步变量的内存地址获知该同步变量被更新后,才能被激活成普通任务,从观察者队列移至执行队列等待执行,以此方式实现任务同步。采用上述任务同步方法能够保证每个生产者 (producer) 和消费者 (consumer) 都能对 n 进行更新。

[0079] 本实施例中,如果 producer\_watcher 已经被激活,producer\_watcher 的激活可能是由 producer\_runner 或者 consumer\_runner 对 m 和 n 的更新而触发的。此时,producer\_watcher 可以根据 m 的值来判断激活操作来源于 producer\_runner 还是 consumer\_runner。

[0080] 具体的,当 m 为 1 时,表明 producer\_runner 提交成功,即 producer\_runner 的执行结果已经将该任务的内存空间快照及其对应的内存空间更新,即此时 producer\_runner 的更新已应用于全局内存。

[0081] 当 m 为 0 时,表明 consumer\_runner 提交成功,即 consumer\_runner 的执行结果已经将该任务的内存空间快照及其对应的内存空间更新。而 producer\_runner 或者还没提交

执行结果,或者提交执行结果时,发现 producer\_runner 的内存空间快照对应的内存空间已经被 consumer\_runner 更新,从而放弃 producer\_runner 本次所有操作及其对内存空间快照及其对应的内存空间的修改。此时,优选的,producer\_watcher 会重新创建 producer\_runner 和 producer\_watcher,以保证 producer\_runner 在提交失败、放弃此任务的全部操作的情况下,也能够再次被调度执行。

[0082] 本发明实施例的方法可以由通用集成电路,如 CPU,或 ASIC 执行。

[0083] 相应的,如图 3 所示,本发明还提供一种用于分布式共享存储的任务同步装置,包括:

[0084] 创建单元 11,用于创建第一任务的内存空间快照和第二任务的同步观察区,所述第二任务的同步观察区包括所述第二任务的同步变量的内存地址,所述同步变量为待同步的任务实现同步所必需的参数;

[0085] 更新单元 12,用于将所述第一任务分配给第一计算节点执行并在所述第一任务执行结束后更新所述第一任务的内存空间快照和所述第一任务的内存空间快照对应的内存空间;

[0086] 同步单元 13,用于当如果所述第二任务的同步变量被所述第一任务更新时,使所述第二任务根据所述第二任务的同步观察区中的同步变量的内存地址获知所述第二任务的同步变量,并将所述第二任务分配给第二计算节点,以使所述第二任务随所述第二任务的同步变量的更新而同步执行。

[0087] 采用上述技术方案后,本发明实施例提供的用于分布式共享存储的任务同步装置,创建了第一任务的内存空间快照和第二任务的同步观察区,所述第二任务的同步观察区存储有所述第二任务的同步变量的内存地址。一方面,第一任务在计算节点本地执行,任务完成后才更新共享存储空间,另一方面,如果所述第二任务的同步变量被所述第一任务更新,可以使所述第二任务根据所述第二任务的同步变量的内存地址获知所述第二任务的同步变量,并将所述第二任务分配给第二计算节点,以使所述第二任务随所述第二任务的同步变量的更新而同步执行,在很好地维护了分布式共享存储系统的内存一致性即程序对内存操作的正确性的同时,大大减少了分布式共享存储的任务同步和内存一致性维护所需要的频繁的消息发送和状态探寻,既简化了编程又有效提高了系统的处理性能。

[0088] 具体的,如图 4 所示,更新单元 12 可包括:

[0089] 第一更新子单元 121,用于如果当所述第一任务的内存空间快照对应的内存空间已经被所述第一任务以外的任务更新时,则放弃对所述第一任务的内存空间快照及其对应的内存空间的更新;

[0090] 第二更新子单元 122,用于当如果所述第一任务的内存空间快照对应的内存空间尚未被所述第一任务以外的任务更新时,则更新所述第一任务的内存空间快照及其对应的内存空间。

[0091] 进一步的,如图 5 所示,在本发明的另一个实施例中,所述装置还包括观察者调度单元 14,用于将所述第二任务放入观察者队列,所述观察者队列为存放具有同步观察区、但不能被立即执行的任务的队列。则,同步单元 13 可包括:

[0092] 第一同步子单元 131,用于当所述第二任务的同步变量被所述第一任务更新时,使所述第二任务根据所述第二任务的同步观察区中的同步变量的内存地址获知所述第二任

务的同步变量；

[0093] 第二同步子单元 132，用于在将所述第二任务从所述观察者队列移至执行队列后，将所述第二任务分配给所述第二计算节点，所述执行队列为存放即将被执行的任务的队列。

[0094] 可选的，第二同步子单元 132，可具体用于在将观察者队列中包括所述第二任务在内的、所有同步变量被所述第一任务更新的任务，从所述观察者队列移至所述执行队列后，将所述第二任务分配给所述第二计算节点。

[0095] 具体的，第二同步子单元 132，可具体用于在将所述第二任务从所述观察者队列移至执行队列后，如果所述第二任务的同步变量中还存在至少一个同步变量所对应的内存空间没有被更新，则将所述第二任务重新放入所述观察者队列；在将所述第二任务从所述观察者队列移至执行队列后，如果所述第二任务的同步变量所对应的内存空间均已被更新，则执行所述第二任务。

[0096] 本发明实施例的方法可以由通用集成电路，如 CPU，或 ASIC 等执行。

[0097] 相应的，本发明还提供一种分布式共享存储系统，包括第一计算节点、第二计算节点以及前述实施例中提供的任一项用于分布式共享存储的任务同步装置。所述用于分布式共享存储的任务同步装置位于所述第一计算节点内，或者位于所述第二计算节点内，或者分别与所述第一计算节点和所述第二计算节点相连，用于将所述第一计算节点与所述第二计算节点中的需要同步的任务进行同步。由于所述分布式共享存储系统中包括有前述实施例中的用于分布式共享存储的任务同步装置，因此也能实现该装置能够实现的有益技术效果，前文已经进行了详细的说明，此处不再赘述。

[0098] 本领域普通技术人员可以理解：实现上述方法实施例的全部或部分流程可以通过计算机程序指令相关的硬件来完成，前述的程序可以存储于一计算机可读取存储介质中，该程序在执行时，执行包括上述方法实施例的步骤；而前述的存储介质包括：ROM、RAM、磁碟或者光盘等各种可以存储程序代码的介质。

[0099] 以上所述，仅为本发明的具体实施方式，但本发明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发明揭露的技术范围内，可轻易想到变化或替换，都应涵盖在本发明的保护范围之内。因此，本发明的保护范围应以所述权利要求的保护范围为准。

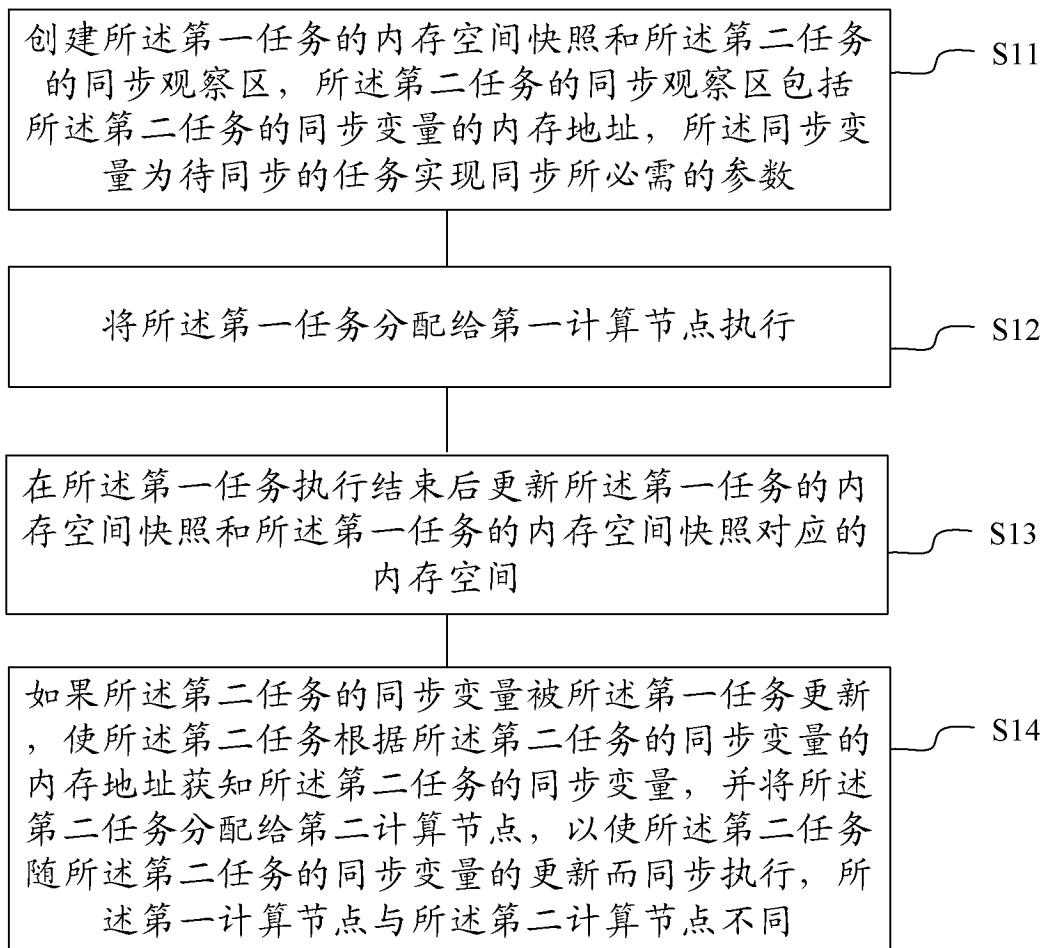


图 1

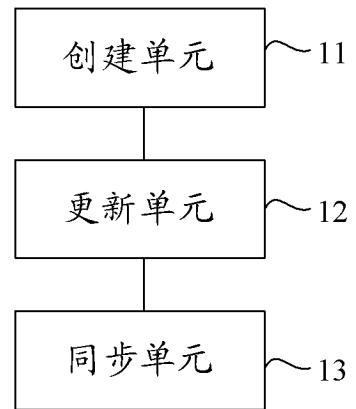
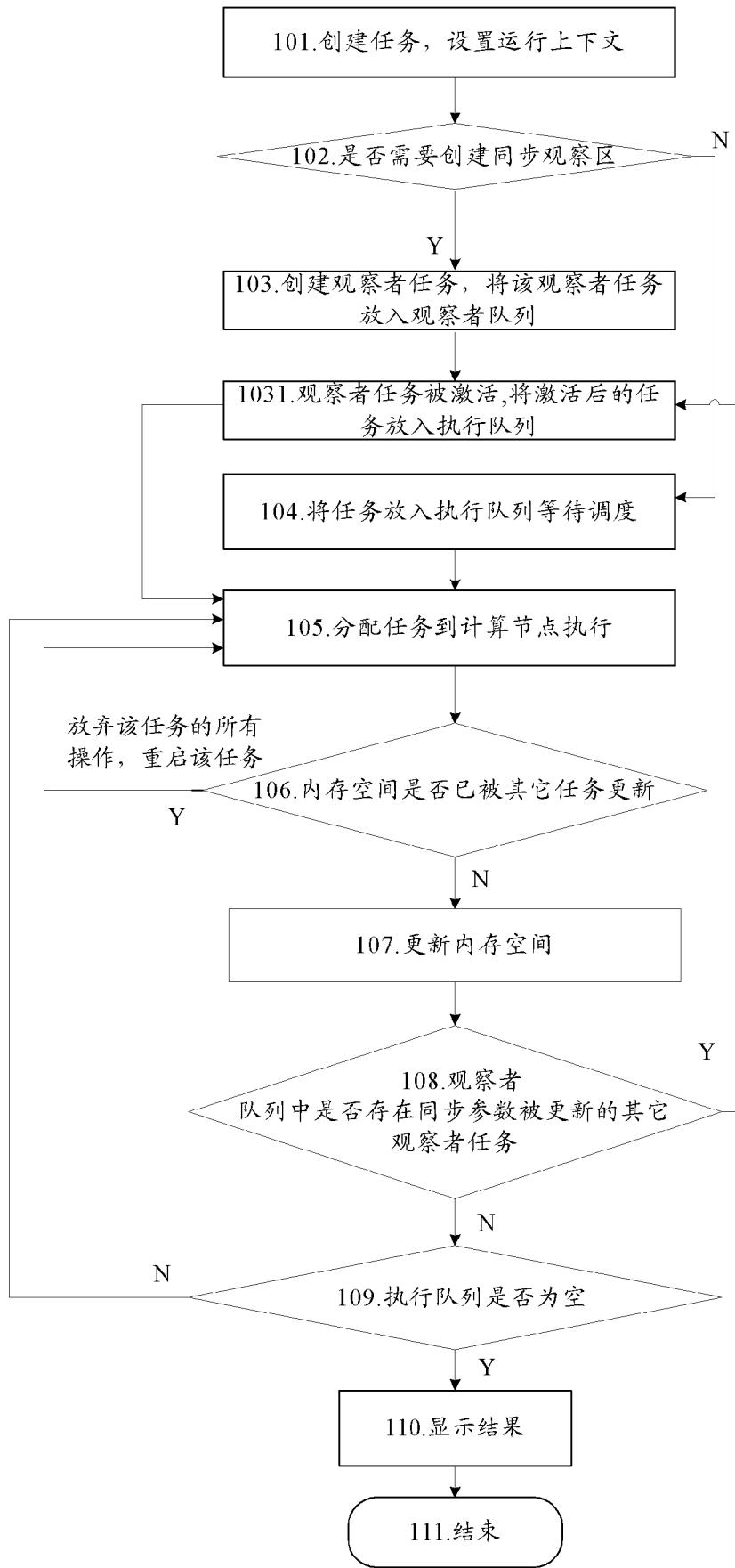


图 3

图 2

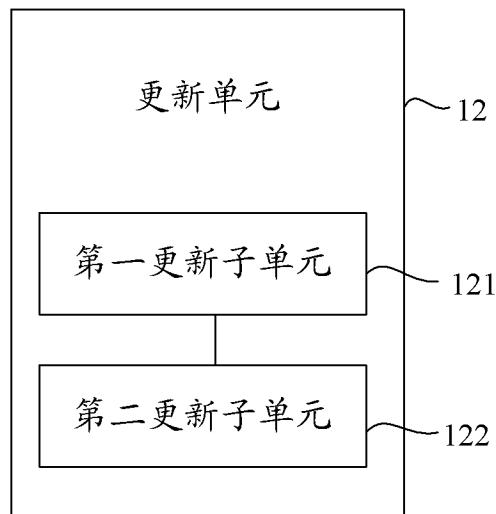


图 4

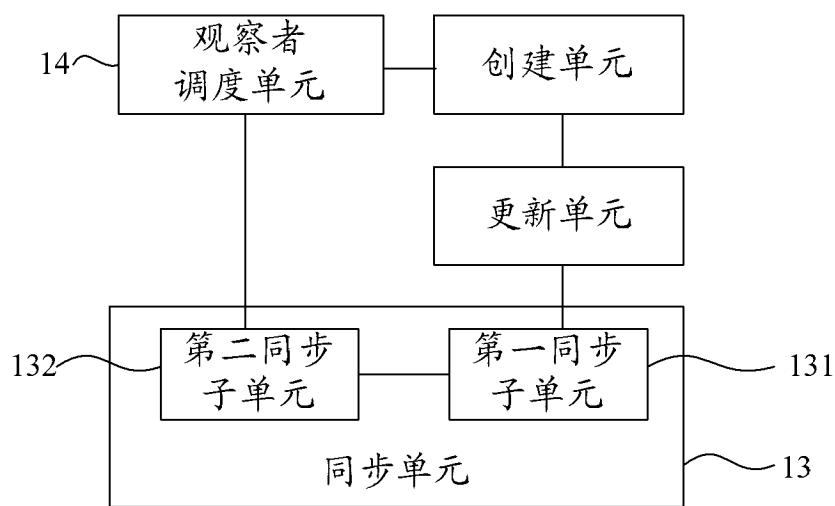


图 5