



(12) 发明专利申请

(10) 申请公布号 CN 103297456 A

(43) 申请公布日 2013. 09. 11

(21) 申请号 201210046660. 4

(22) 申请日 2012. 02. 24

(71) 申请人 阿里巴巴集团控股有限公司
地址 英属开曼群岛大开曼岛资本大厦一座
四层 847 号邮箱

(72) 发明人 舒宏旺

(74) 专利代理机构 北京同达信恒知识产权代理
有限公司 11291
代理人 郭润湘

(51) Int. Cl.
H04L 29/08 (2006. 01)
H04L 29/06 (2006. 01)

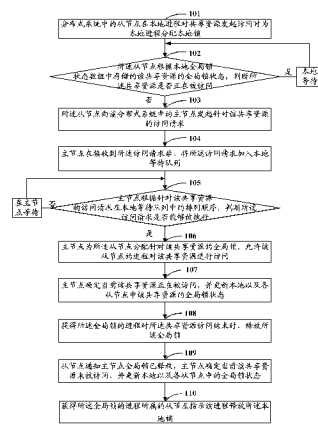
权利要求书2页 说明书9页 附图3页

(54) 发明名称

一种分布式系统下共享资源的访问方法及分布式系统

(57) 摘要

本申请公开了一种分布式系统下共享资源的访问方法及分布式系统,所述分布式系统中的从节点在本地进程对共享资源发起访问时,根据本地全局锁状态数组中存储的该共享资源的全局锁状态,判断出所述共享资源未正在被访问时,向分布式系统中的主节点发起针对该共享资源的访问请求,在主节点根据针对该共享资源的各访问请求在本地等待队列中的排列顺序,判断所述从节点的访问请求能够被执行时,获取该共享资源的全局锁;通过本申请的技术方案,所述分布式系统可以利用系统本身的锁机制来实现整个系统的同步、互斥控制,无需通过第三方集中式系统提供锁服务来实现,从而减少了对外界集中式锁服务系统的依赖,降低了系统的风险。



1. 一种分布式系统下共享资源的访问方法,其特征在于,所述方法包括:

分布式系统中的从节点在本地进程对共享资源发起访问时,根据本地全局锁状态数组中存储的该共享资源的全局锁状态,判断出所述共享资源未正在被访问时,向该分布式系统中的主节点发起针对该共享资源的访问请求;

主节点在接收到所述访问请求后,将所述访问请求加入本地等待队列,并根据针对该共享资源的访问请求在本地等待队列中的排列顺序,判断出所述访问请求能够被执行时,为所述从节点分配针对该共享资源的全局锁,允许该从节点的进程对该共享资源进行访问。

2. 如权利要求 1 所述的共享资源的访问方法,其特征在于,在从节点的本地进程对该共享资源发起访问之后,从节点判断所述共享资源是否正在被访问之前,所述方法还包括:

从节点将对该共享资源发起访问的本地的多个进程进行排序,并为排列在最前的进程分配本地锁;

当从节点获得主节点分配的针对该共享资源的全局锁时,所述方法还包括:

从节点指示当前分配了本地锁的进程对该共享资源进行访问,并在访问结束时,指示该进程释放所述全局锁和本地锁;

从节点从排序后的未被分配本地锁的对该共享资源发起访问的本地进程中,为排列在最前的进程分配本地锁。

3. 如权利要求 1 所述的共享资源的访问方法,其特征在于,所述方法还包括:

主节点在判断出针对该共享资源的访问请求不能被执行后,为本地等待队列中能够被执行的其他访问请求对应的从节点分配针对该共享资源的全局锁,允许所述其他访问请求对应的从节点的进程对该共享资源进行访问;

当获得该共享资源的全局锁的从节点的进程完成对该共享资源的访问后,所述方法还包括:

主节点删除本地等待队列中获得该共享资源的全局锁的从节点的访问请求。

4. 如权利要求 2 或 3 所述的共享资源的访问方法,其特征在于,主节点为从节点分配针对该共享资源的全局锁之后,所述方法还包括:

主节点确定当前该共享资源正在被访问,并更新本地以及分布式系统中各从节点中的全局锁状态数组中存储的该共享资源的全局锁状态。

5. 如权利要求 2 所述的共享资源的访问方法,其特征在于,进程释放所述全局锁之后,所述方法还包括:

释放所述全局锁的进程所属的从节点通知主节点所述全局锁已释放;

主节点确定当前该共享资源未被访问,并更新本地以及分布式系统中各从节点中的全局锁状态数组中存储的该共享资源的全局锁状态。

6. 一种分布式系统,其特征在于,所述分布式系统包括:

从节点,用于在本地进程对共享资源发起访问时,根据本地全局锁状态数组中存储的该共享资源的全局锁状态,判断出所述共享资源未正在被访问时,向主节点发起针对该共享资源的访问请求;

主节点,用于在接收到从节点发起的所述访问请求后,将所述访问请求加入本地等待

队列,并根据针对该共享资源的访问请求在本地等待队列中的排列顺序,判断出所述访问请求能够被执行时,为所述从节点分配针对该共享资源的全局锁,允许该从节点的进程对该共享资源进行访问。

7. 如权利要求 6 所述的分布式系统,其特征在于,

所述从节点,还用于在本地进程对该共享资源发起访问之后,从节点判断所述共享资源是否正在被访问之前,将对该共享资源发起访问的本地的多个进程进行排序,并为排列在最前的进程分配本地锁,以及,在获得主节点分配的针对该共享资源的全局锁时,指示当前分配了本地锁的进程对该共享资源进行访问,并在访问结束时,指示该进程释放所述全局锁和本地锁,并从排序后的未被分配本地锁的对该共享资源发起访问的本地进程中,为排列在最前的进程分配本地锁。

8. 如权利要求 6 所述的分布式系统,其特征在于,

所述主节点,还用于在判断出针对该共享资源的访问请求不能被执行后,为本地等待队列中能够被执行的其他访问请求对应的从节点分配针对该共享资源的全局锁,允许所述能够被执行的其他访问请求对应的从节点的进程对该共享资源进行访问;以及,在获得该共享资源的全局锁的从节点的进程完成对该共享资源的访问后,删除本地等待队列中获得该共享资源的全局锁的从节点的访问请求。

9. 如权利要求 7 或 8 所述的分布式系统,其特征在于,

所述主节点,还用于在为从节点分配针对该共享资源的全局锁之后,确定当前该共享资源正在被访问,并更新本地以及各从节点中的全局锁状态数组中存储的该共享资源的全局锁状态。

10. 如权利要求 7 所述的分布式系统,其特征在于,

所述从节点,还用于在获取到全局锁的本地进程释放所述全局锁之后,通知主节点所述全局锁已释放;

所述主节点,还用于在接收到所述从节点释放全局锁的通知后,确定当前该共享资源未被访问,并更新本地以及各从节点中的全局锁状态数组中存储的该共享资源的全局锁状态。

一种分布式系统下共享资源的访问方法及分布式系统

技术领域

[0001] 本申请涉及分布式系统领域,尤其涉及一种分布式系统下共享资源的访问方法及分布式系统。

背景技术

[0002] 分布式系统是由地理上和物理上分散的各自独立的多个节点(如服务器、处理单元等)组成的系统,所述多个节点连接在一起以实现对不同资源的共享,一个分布式系统相当于一个统一的整体。

[0003] 对于分布式系统来说,系统内的多个节点在并行运行,在同一时刻可能存在多个节点或同一节点的多个进程对同一共享资源发起并发访问,若不对其进行控制,将会导致各种定时相关性和非确定性行为不受控制,导致反常状况的发生,如共享资源数据破坏、进程损毁等。因此,需要对多个进程对同一共享资源的访问顺序进行控制,以便实现多个进程对该共享资源的有序化访问,保证分布式系统的正常运行。

[0004] 目前,锁是实现并发访问控制的常用手段,对于单个节点来说,当同一时刻,该节点的多个进程针对某一共享资源发起并发访问时,可以为该共享资源设置一个锁,通过进程进入临界区时的关锁和退出临界区时的开锁来达到该节点的多进程对该共享资源的有序化访问,实现单个节点系统的同步、互斥控制。但是,由于分布式系统具有多个节点,各节点具有各自独立的物理内存空间,并没有一个统一的可共享的主存,因此,不能通过单个节点自身的锁来实现整个分布式系统的同步、互斥控制。

[0005] 针对于此,业界通常采用外部集中式的锁服务系统来解决分布式系统的同步、互斥问题,如图1所示,其为基于锁服务系统实现分布式系统同步、互斥控制的系统结构示意图。所述系统包括锁服务系统、分布式系统,所述分布式系统中包含多个节点,如节点1、节点2以及节点3。

[0006] 所述系统的工作原理为:当某一时刻,节点1和节点2的进程均向共享资源A发起访问时,节点1和节点2的进程需向系统中的锁服务系统发送请求报文来申请获得该共享资源A的锁;锁服务系统在接收到节点1和节点2的进程的请求报文后,根据目前共享资源A的访问状态,同意或拒绝节点1或节点2的进程对共享资源A的访问请求。

[0007] 例如,若锁服务系统在接收到节点1的进程的请求报文后,确定当前并没有进程在对共享资源A进行访问,则同意节点1的进程获得所述共享资源A的锁,节点1的进程可对共享资源A进行访问。

[0008] 若锁服务系统在接收到节点1的进程的请求报文后,确定当前节点2的进程正在占用共享资源A的锁,对共享资源A进行访问时,则拒绝节点1的进程对共享资源A的访问请求,并将其请求报文存放在缓冲队列中,待节点2的进程完成访问、释放所述共享资源A的锁后,才同意节点1的进程获得所述共享资源A的锁并对共享资源A进行访问,从而保证同一时刻只有一个进程可以成功获得共享资源A的锁,实现对共享资源A的有序性访问。

[0009] 在上述基于锁服务系统来实现分布式系统同步、互斥控制的系统中,虽然可以控

制多个节点的多个进程对同一共享资源的并发访问,解决整个分布式系统的同步、互斥问题,但是,所述外部集中式的锁服务系统结构复杂、不易搭建、需要长期的资源来维护,且对所述外部锁服务系统的依赖会对分布式系统本身的安全性造成一定的风险。

发明内容

[0010] 本申请的目的在于,提供一种分布式系统下共享资源的访问方法及分布式系统,用以解决现有技术中存在的依赖外部锁服务系统来实现分布式系统的同步、互斥控制,所造成的系统风险较高的问题。

[0011] 一种分布式系统下共享资源的访问方法,所述方法包括:

[0012] 分布式系统中的从节点在本地进程对共享资源发起访问时,根据本地全局锁状态数组中存储的该共享资源的全局锁状态,判断出所述共享资源未正在被访问时,向该分布式系统中的主节点发起针对该共享资源的访问请求;

[0013] 主节点在接收到所述访问请求后,将所述访问请求加入本地等待队列,并根据针对该共享资源的访问请求在本地等待队列中的排列顺序,判断出所述访问请求能够被执行时,为所述从节点分配针对该共享资源的全局锁,允许该从节点的进程对该共享资源进行访问。

[0014] 一种分布式系统,所述分布式系统包括:

[0015] 从节点,用于在本地进程对共享资源发起访问时,根据本地全局锁状态数组中存储的该共享资源的全局锁状态,判断出所述共享资源未正在被访问时,向主节点发起针对该共享资源的访问请求;

[0016] 主节点,用于在接收到从节点发起的所述访问请求后,将所述访问请求加入本地等待队列,并根据针对该共享资源的访问请求在本地等待队列中的排列顺序,判断出所述访问请求能够被执行时,为所述从节点分配针对该共享资源的全局锁,允许该从节点的进程对该共享资源进行访问。

[0017] 本申请的有益效果如下:

[0018] 本申请实施例提供了一种分布式系统下共享资源的访问方法及分布式系统,所述分布式系统中的从节点在本地进程对共享资源发起访问时,根据本地全局锁状态数组中存储的该共享资源的全局锁状态,判断所述共享资源未被访问时,向分布式系统中的主节点发起针对该共享资源的访问请求,并在主节点根据针对该共享资源的访问请求在本地等待队列中的排列顺序,确定可以执行所述访问请求时,获取该共享资源的全局锁,并对该共享资源进行访问。通过本申请的技术方案,所述分布式系统可通过系统本身的从节点和主节点来分别判断是否向主节点发起针对某共享资源的访问请求以及是否允许该从节点获取所述共享资源的全局锁,实现对该共享资源的有序访问,无需利用外界锁服务系统提供的锁服务来解决系统的同步、互斥问题,降低了对外界锁服务系统的依赖造成的系统安全性风险。

附图说明

[0019] 图 1 所示为基于锁服务系统实现分布式系统同步、互斥控制的系统结构示意图;

[0020] 图 2 所示为本申请实施例一中分布式系统下共享资源的访问方法流程示意图;

[0021] 图 3 所示为本申请实施例二中分布式系统的结构示意图。

具体实施方式

[0022] 本申请实施例提供了一种分布式系统下共享资源的访问方法及分布式系统,在分布式系统中的多个进程对同一共享资源发起访问时,为了避免多个进程并发争用同一共享资源导致的共享资源数据一致性破坏或进程损毁等异常状况,所述分布式系统中的从节点在本地进程对共享资源发起访问时,需要根据本地全局锁状态数组中存储的该共享资源的全局锁状态,判断所述共享资源是否正在被访问,若否,则向分布式系统中的主节点发起针对该共享资源的访问请求,并在主节点根据针对该共享资源的访问请求在本地等待队列中的排列顺序,确定可以执行所述访问请求时,获取该共享资源的全局锁,实现对该共享资源的访问,无需利用外界锁服务系统提供的锁服务来解决系统的同步、互斥问题,降低了对外界锁服务系统的依赖造成的系统安全性风险。

[0023] 实施例一:

[0024] 本申请实施例一提供了一种分布式系统下共享资源的访问方法,所述方法可以应用在具有多个节点的分布式系统中。所述分布式系统在运行时会从各节点中选取一个主节点,其它节点成为从节点,主节点与从节点之间可通过传输控制协议(Transmission Control Protocol, TCP)长连接或 TCP 短连接的方式保持通信。

[0025] 具体地,所述分布式系统具体可以通过Fast Paxos等算法来实现主节点的选举以及分布式系统中数据的一致性。

[0026] 优选地,所述分布式系统中的各节点的结构均相同,均包括:本地锁、全局锁状态数组、Log(日志)记录以及等待队列四种结构组件。

[0027] 所述本地锁为节点自身的锁,节点中的各进程通过争用该节点的本地锁,以得到在整个分布式系统中与其他各节点中的进程争用全局锁,从而访问整个分布式系统中统一的共享资源的权利。

[0028] 所述全局锁状态数组用来存储分布式系统中各共享资源的全局锁的状态,还可用来存储各共享资源的全局锁的标识以及各共享资源的标识等信息。由于分布式系统中有大量的共享资源,每个共享资源均有自己的全局锁,因此,系统中的全局锁并不唯一,在此情况下,本申请实施例中通过在各节点中设置全局锁状态数组,来以数组的方式存储各共享资源的全局锁状态。各共享资源的全局锁状态对应全局锁状态数组中的一个元素值,例如,若针对某一共享资源来说,当前有进程对其进行访问,则可在全局锁状态数组中设置该共享资源的全局锁状态为 1,代表该共享资源的全局锁状态为被占用;若当前并未有进程对其进行访问,则可在全局锁状态数组中设置该共享资源的全局锁状态为 0,代表该共享资源的全局锁状态为未被占用。

[0029] 所述 Log 记录,用来保存系统中各节点的操作处理记录,以便将来做为参考,在各节点宕机后恢复时使用;本申请实施例中,可以采用快照信息的方式来保存所述 Log 记录,从而减小有用日志的大小,快照时间根据需要设置。

[0030] 所述等待队列,用于进行消息的串行化,主要应用于主节点;当各从节点针对某一共享资源,向主节点发送访问请求时,所述访问请求将以队列的方式存储在主节点的等待队列中。

[0031] 如图 2 所示,其为本申请实施例一中分布式系统下共享资源的访问方法流程示意图,所述方法包括以下步骤:

[0032] 步骤 101:分布式系统中的从节点在本地进程对共享资源发起访问时,为本地进程分配本地锁。

[0033] 本步骤 101 中,所述从节点根据该节点应用自身的机制为该节点的本地进程分配本地锁。具体地,所述从节点可以将对该共享资源发起访问的本地的多个进程进行排序,并为排列在最前的进程分配本地锁。

[0034] 进一步地,所述从节点可以按照对共享资源发起访问的本地多个进程的优先级高低、发起访问的时间先后顺序以及随机性原则或其他排序原则,对各进程进行排序,并为排列在最前的进程分配本地锁,本申请实施例对此并不进行限定。

[0035] 在执行完步骤 101 后,在同一时刻,所述从节点中对该共享资源发起访问的本地的多个进程中,将只有一个进程获得本地锁,得到在整个分布式系统中与其他各节点中的进程争用该共享资源的全局锁、从而访问该共享资源的权利,实现该从节点的同步。

[0036] 步骤 102:所述从节点根据本地全局锁状态数组中存储的该共享资源的全局锁状态,判断所述共享资源是否正在被访问,若是,则在本地等待,并在设定时长内继续执行步骤 102 的判断过程;若否,则执行步骤 103。

[0037] 具体地,若对进程的执行响应速率要求较高,则设定较短的等待时长,若对进程的执行响应速率要求较低,可设定较长的等待时长。实际操作中,所述设定时长一般均较短,以保证系统较高的处理速率。

[0038] 特殊地,当从节点与主节点失去联系时,可通过设置该从节点本地全局锁状态数组的方式来实现,如将该从节点本地全局锁状态数组中存储的各共享资源的全局锁状态均设置为被占用的状态,此种情况下,所述从节点无需执行步骤 103,即与主节点失去联系。

[0039] 步骤 103:所述从节点向该分布式系统中的主节点发起针对该共享资源的访问请求。

[0040] 所述访问请求中携带该共享资源的标识以及该从节点的标识等信息。

[0041] 具体地,在本申请实施例所述共享资源的访问方法中,由于从节点在向分布式系统中的主节点发起针对某一共享资源的访问请求之前,该从节点的本地进程中对该共享资源发起访问的多个进程已进行了本地锁的争用,在同一时刻,所述从节点中只有一个进程获得本地锁,因此,可以避免同一时刻,该从节点的多个进程(包括获取到本地锁的进程以及其他未获取到本地锁的进程)均向主节点发起针对该共享资源的访问请求,这降低了所述从节点对主节点访问请求的次数,使得主节点在同一时刻接收到相对较少的访问请求,提高了系统的处理效率。

[0042] 步骤 104:主节点在接收到所述访问请求后,将所述访问请求加入本地等待队列。

[0043] 具体地,主节点可以按照从节点发起访问请求的时间先后顺序,将所述从节点的访问请求加入到本地等待队列,也可以按照从节点对共享资源发起访问的优先级高低、随机性原则或其他排序方式将所述从节点的访问请求加入到本地等待队列;进一步地,所述主节点可以按照实际需要,随时更新本地等待队列。

[0044] 步骤 105:主节点根据针对该共享资源的访问请求在本地等待队列中的排列顺序,判断所述访问请求是否能够被执行,若是,则执行步骤 106,否则,指示所述访问请求在

主节点的本地等待队列等待,并在设定时长内,继续执行步骤 105 的判断过程。

[0045] 具体地,主节点可以按照各共享资源的标识,在本地等待队列中分别为针对各共享资源发起的访问请求进行排序,也就是说,针对任一共享资源,本地等待队列中都有相应的针对该共享资源发起的访问请求的排序队列。

[0046] 主节点可以根据针对该共享资源的访问请求在本地等待队列中的排列顺序,判断所述访问请求是否排列在最前,若是,则表示所述访问请求能够被执行。

[0047] 由于主节点与各从节点之间通过 TCP 长连接或 TCP 短连接的方式进行消息的通讯,存在一定的传输时延,当从节点根据本地全局锁状态数组中存储的该共享资源的全局锁状态确定所述共享资源未被访问,从而向主节点发送访问请求时,主节点可能已经接收到其他从节点的访问请求,并允许其他从节点对该共享资源进行访问,因此,主节点本地全局锁状态数组中存储的该共享资源的全局锁状态与从节点中的并不一定相同,在此情况下,本申请实施例中主节点在将所述访问请求加入本地等待队列之后,且根据针对该共享资源的访问请求在本地等待队列中的排列顺序判断所述访问请求能否被执行之前,还可以根据本地全局锁状态数组中存储的该共享资源的全局锁状态判断所述共享资源是否正在被访问,若确定该共享资源没有被访问时,再执行本步骤 104 中的判断过程。

[0048] 进一步地,在步骤 105 中,主节点在判断出针对该共享资源的访问请求不能被执行后,还需要为本地等待队列中能够被执行的其他访问请求对应的从节点分配针对该共享资源的全局锁,允许所述其他访问请求对应的从节点的进程对该共享资源进行访问;具体地,所述主节点需要为本地等待队列中排序最前的访问请求对应的从节点分配针对该共享资源的全局锁。

[0049] 步骤 106:主节点为所述从节点分配针对该共享资源的全局锁,允许该从节点的进程对该共享资源进行访问。

[0050] 本申请实施例中,从节点本地全局锁状态数组中的该共享资源的状态为未被占用时,并不表示该从节点可以获得该共享资源的全局锁;所述从节点还需要向主节点发起访问请求,以及到主节点排队等待,直到主节点通知该从节点可以获得所述共享资源的全局锁。

[0051] 具体地,从节点在获取到该共享资源的全局锁后,指示针对该共享资源发起访问的本地进程中当前分配了本地锁的进程对该共享资源进行访问。

[0052] 进一步地,当获得该共享资源的全局锁的从节点的进程完成对该共享资源的访问后,主节点将删除本地等待队列中获得该共享资源的全局锁的从节点的访问请求,以实现依次执行本地等待队列中针对该共享资源的各访问请求的目的。

[0053] 步骤 107:主节点确定当前该共享资源正在被访问,并更新本地以及分布式系统中各从节点中的全局锁状态数组中存储的该共享资源的全局锁状态。

[0054] 为了保证分布式系统中各节点的全局锁状态数组中存储的全局锁状态的及时更新、提高通信效率,主节点与各从节点优先选择 TCP 长连接的连接方式。

[0055] 具体地,主节点可以通过向分布式系统中各从节点发送更新请求消息的方式更新各从节点全局锁状态数组中存储的该共享资源的全局锁状态,相应地,各从节点可以通过向主节点返回更新响应消息的方式通知主节点所述共享资源的全局锁状态已更新。

[0056] 进一步地,主节点可以在确定从节点的访问请求能够被执行之后,且执行为该从

节点分配该共享资源的全局锁之前或同时,更新本地以及分布式系统中各从节点中的全局锁状态数组中存储的该共享资源的全局锁状态,以避免出现共享资源正在被访问而该共享资源的全局锁状态仍为未被占用的情况。

[0057] 步骤 108:获得所述全局锁的进程所属的从节点在获得本地锁的进程对所述共享资源访问结束时,指示该进程释放所述全局锁。

[0058] 获得所述全局锁的进程所属的从节点中获得本地锁的进程,利用获取到的全局锁对该共享资源进行访问处理,并在访问结束时,释放所述全局锁。

[0059] 步骤 109:释放所述全局锁的进程所属的从节点通知主节点所述全局锁已释放,主节点确定当前该共享资源未被访问,并更新本地以及分布式系统中各从节点中的全局锁状态数组中存储的该共享资源的全局锁状态。

[0060] 具体地,所述从节点可以通过向主节点发送释放全局锁请求报文的方式或其他方式通知主节点所述全局锁已释放;主节点在接收到所述从节点发送的释放全局锁请求报文后,确定当前该共享资源未被访问,在更新本地全局锁状态数组中存储的该共享资源的全局锁状态后,及时通知其他各节点,实现全局锁状态的实时更新。

[0061] 步骤 110:获得所述全局锁的进程所属的从节点指示该进程释放所述本地锁。

[0062] 具体地,所述从节点在指示所述进程释放本地锁后,从排序后的未被分配本地锁针对该共享资源发起访问的本地进程中,为排列在最前的进程分配本地锁。

[0063] 需要说明的是,获得所述全局锁的进程所属的从节点可以在获得本地锁的进程对所述共享资源访问结束时,指示该进程同时释放所述全局锁和本地锁,并通知主节点所述全局锁已释放,之后,主节点确定当前该共享资源未被访问,并更新本地以及分布式系统中各从节点中的全局锁状态数组中存储的该共享资源的全局锁状态。

[0064] 需要说明的是,在本申请实施例一中,主节点不但可以作为协调者来对各从节点是否能够得到共享资源的全局锁,从而对该共享资源进行访问进行判断,其本身也可以对分布式系统中的各共享资源进行访问,并按照与本申请实施例一类似的过程完成获取共享资源的全局锁、实现对该共享资源进行访问的目的。

[0065] 通过本申请实施例一的方案,所述分布式系统可以通过引用 library 库、调用 API 的方式获取、释放各共享资源的全局锁,实现对分布式系统中各共享资源的有序化访问,可以实现分布式系统的同步、互斥控制,这避免了分布式系统中多个进程并发争用同一共享资源导致的共享资源数据一致性破坏或进程损毁等异常状况,并且无需利用外界锁服务系统,降低了系统安全性风险并且使用方便。

[0066] 实施例二:

[0067] 如图 3 所示,其为本申请实施例二中分布式系统的结构示意图,所述分布式系统包括一个主节点 31 和至少一个从节点 32,各节点均包括本地锁、全局锁状态数组、Log 记录以及等待队列四种结构组件。

[0068] 具体地,所述分布式系统可以通过 Fast Paxos 等算法来实现主节点 31 的选举以及分布式系统中数据的一致性,主节点 31 与从节点 32 之间通过 TCP 长连接或 TCP 短连接的方式保持通信。

[0069] 所述从节点 32,用于在本地进程对共享资源发起访问时,根据本地全局锁状态数组中存储的该共享资源的全局锁状态,判断所述共享资源是否正在被访问,在判断出所述

共享资源未正在被访问时,向主节点 31 发起针对该共享资源的访问请求。

[0070] 所述主节点 31,用于在接收到从节点 32 发起的所述访问请求后,将所述访问请求加入本地等待队列,并根据针对该共享资源的访问请求在本地等待队列中的排列顺序,判断所述访问请求是否能够被执行,若判断出所述访问请求能够被执行时,则为所述从节点 32 分配针对该共享资源的全局锁,允许该从节点 32 的进程对该共享资源进行访问,否则,指示所述访问请求在本地等待队列等待。

[0071] 由于主节点 31 与各从节点 32 之间通过 TCP 长连接或 TCP 短连接的方式进行消息的通讯,存在一定的传输时延,当从节点 32 根据本地全局锁状态数组中存储的该共享资源的全局锁状态确定所述共享资源未被访问,从而向主节点 31 发送访问请求时,主节点 31 可能已经接收到其他从节点 32 的访问请求,并允许其他从节点 32 对该共享资源进行访问,因此,主节点 31 本地全局锁状态数组中存储的该共享资源的全局锁状态与从节点 32 中的并不一定相同,在此情况下,本申请实施例中主节点 31 在将所述访问请求加入本地等待队列之后,且根据针对该共享资源的访问请求在本地等待队列中的排列顺序判断所述访问请求能否被执行之前,还可以根据本地全局锁状态数组中存储的该共享资源的全局锁状态判断所述共享资源是否正在被访问,若确定该共享资源没有被访问时,再根据针对该共享资源的访问请求在本地等待队列中的排列顺序判断所述访问请求能否被执行。

[0072] 优选地,所述从节点 32 还用于在本地进程对该共享资源发起访问之后,自身判断所述共享资源是否正在被访问之前,将对所述共享资源发起访问的本地的多个进程进行排序,并为排列在最前的进程分配本地锁,以及,在获得主节点 31 分配的针对该共享资源的全局锁时,指示当前分配了本地锁的进程对该共享资源进行访问,并在访问结束时,指示该进程释放所述全局锁和本地锁,并从排序后的未被分配本地锁的对该共享资源发起访问的本地进程中,为排列在最前的进程分配本地锁。

[0073] 由于从节点 32 在向分布式系统中的主节点 31 发起针对某一共享资源的访问请求之前,该从节点 32 的本地进程中对所述共享资源发起访问的多个进程已进行了本地锁的争用,在同一时刻,所述从节点 32 只有一个进程获得本地锁,因此,可以避免同一时刻,该从节点 32 的多个进程均向主节点 31 发起针对该共享资源的访问请求,降低所述从节点 32 对主节点 31 访问请求的次数,使得主节点 31 在同一时刻接收到相对较少的访问请求,提高系统的处理效率。

[0074] 进一步地,所述主节点 31 还用于在判断出针对该共享资源的访问请求不能被执行后,为本地等待队列中能够被执行的其他访问请求对应的从节点 32 分配针对该共享资源的全局锁,允许所述能够被执行的其他访问请求对应的从节点 32 的进程对该共享资源进行访问;以及,在获得该共享资源的全局锁的从节点 32 的进程完成对该共享资源的访问后,删除本地等待队列中获得该共享资源的全局锁的从节点 32 的访问请求。

[0075] 当主节点 31 接收到某从节点 32 的访问请求,且确定所述共享资源未被访问时,在同一时刻,可能已有其他从节点 32 向主节点 31 发起针对该共享资源的访问请求,并且依照排序原则,排列在该从节点 32 的访问请求之前,则此时该从节点 32 不能够获得该共享资源的全局锁,还需要到主节点 31 排队等待,直到主节点 31 通知该从节点 32 可以获得所述共享资源的全局锁。

[0076] 进一步地,所述主节点 31 还用于在为从节点 32 分配针对该共享资源的全局锁之

后,确定当前该共享资源正在被访问,并更新本地以及各从节点 32 中的全局锁状态数组中存储的该共享资源的全局锁状态。

[0077] 进一步地,主节点 31 还可以在确定从节点 32 的访问请求能够被执行之后,且执行为该从节点 32 分配该共享资源的全局锁之前或同时,更新本地以及分布式系统中各从节点 32 中的全局锁状态数组中存储的该共享资源的全局锁状态,以避免出现共享资源正在被访问、而该共享资源的全局锁状态仍为未被占用的情况。

[0078] 所述从节点 32,还用于在获取到全局锁的本地进程释放所述全局锁之后,通知主节点 31 所述全局锁已释放;具体地,所述从节点 32 可以通过向主节点 31 发送释放全局锁请求报文的方式或其他方式通知主节点 31 所述全局锁已释放。

[0079] 所述主节点 31 还用于在接收到所述从节点 32 释放全局锁的通知后,确定当前该共享资源未被访问,并更新本地以及各从节点 32 中的全局锁状态数组中存储的该共享资源的全局锁状态;具体地,主节点 31 在接收到所述从节点 32 发送的释放全局锁请求报文后,确定当前该共享资源未被访问,并在更新本地全局锁状态数组中存储的该共享资源的全局锁状态后,及时通知其他各从节点 32,实现全局锁状态的实时更新。

[0080] 需要说明的是,在本申请各实施例中,主节点 31 不但可以作为协调者来对各从节点 32 是否能够得到共享资源的全局锁,从而对该共享资源进行访问进行判断,其本身也可以对分布式系统中的各共享资源进行访问,并按照与本申请实施例一类似的过程完成获取共享资源的全局锁、实现对该共享资源进行访问的目的。

[0081] 本领域内的技术人员应明白,本申请的实施例可提供为方法、系统、或计算机程序产品。因此,本申请可采用完全硬件实施例、完全软件实施例、或结合软件和硬件方面的实施例的形式。而且,本申请可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储介质(包括但不限于磁盘存储器、CD-ROM、光学存储器等)上实施的计算机程序产品的形式。

[0082] 本申请是参照根据本申请实施例的方法、设备(系统)、和计算机程序产品的流程图和/或方框图来描述的。应理解可由计算机程序指令实现流程图和/或方框图中的每一流程和/或方框、以及流程图和/或方框图中的流程和/或方框的结合。可提供这些计算机程序指令到通用计算机、专用计算机、嵌入式处理机或其他可编程数据处理设备的处理器以产生一个机器,使得通过计算机或其他可编程数据处理设备的处理器执行的指令产生用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的装置。

[0083] 这些计算机程序指令也可存储在能引导计算机或其他可编程数据处理设备以特定方式工作的计算机可读存储器中,使得存储在该计算机可读存储器中的指令产生包括指令装置的制品,该指令装置实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能。

[0084] 这些计算机程序指令也可装载到计算机或其他可编程数据处理设备上,使得在计算机或其他可编程设备上执行一系列操作步骤以产生计算机实现的处理,从而在计算机或其他可编程设备上执行的指令提供用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的步骤。

[0085] 尽管已描述了本申请的优选实施例,但本领域内的技术人员一旦得知了基本创造

性概念,则可对这些实施例做出另外的变更和修改。所以,所附权利要求意欲解释为包括优选实施例以及落入本申请范围的所有变更和修改。

[0086] 显然,本领域的技术人员可以对本申请进行各种改动和变型而不脱离本申请的精神和范围。这样,倘若本申请的这些修改和变型属于本申请权利要求及其等同技术的范围之内,则本申请也意图包含这些改动和变型在内。

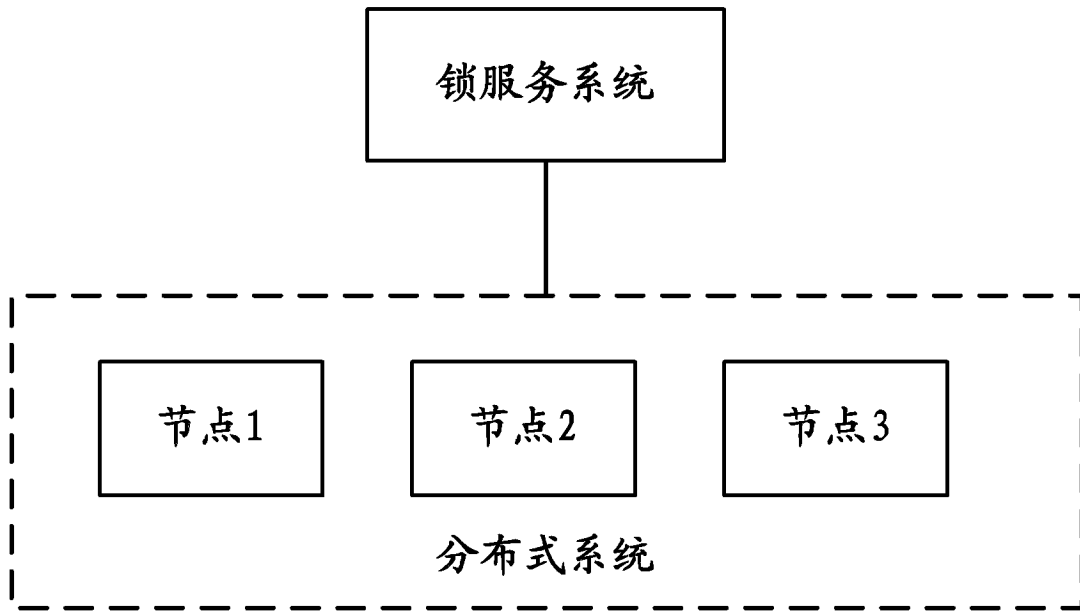


图 1

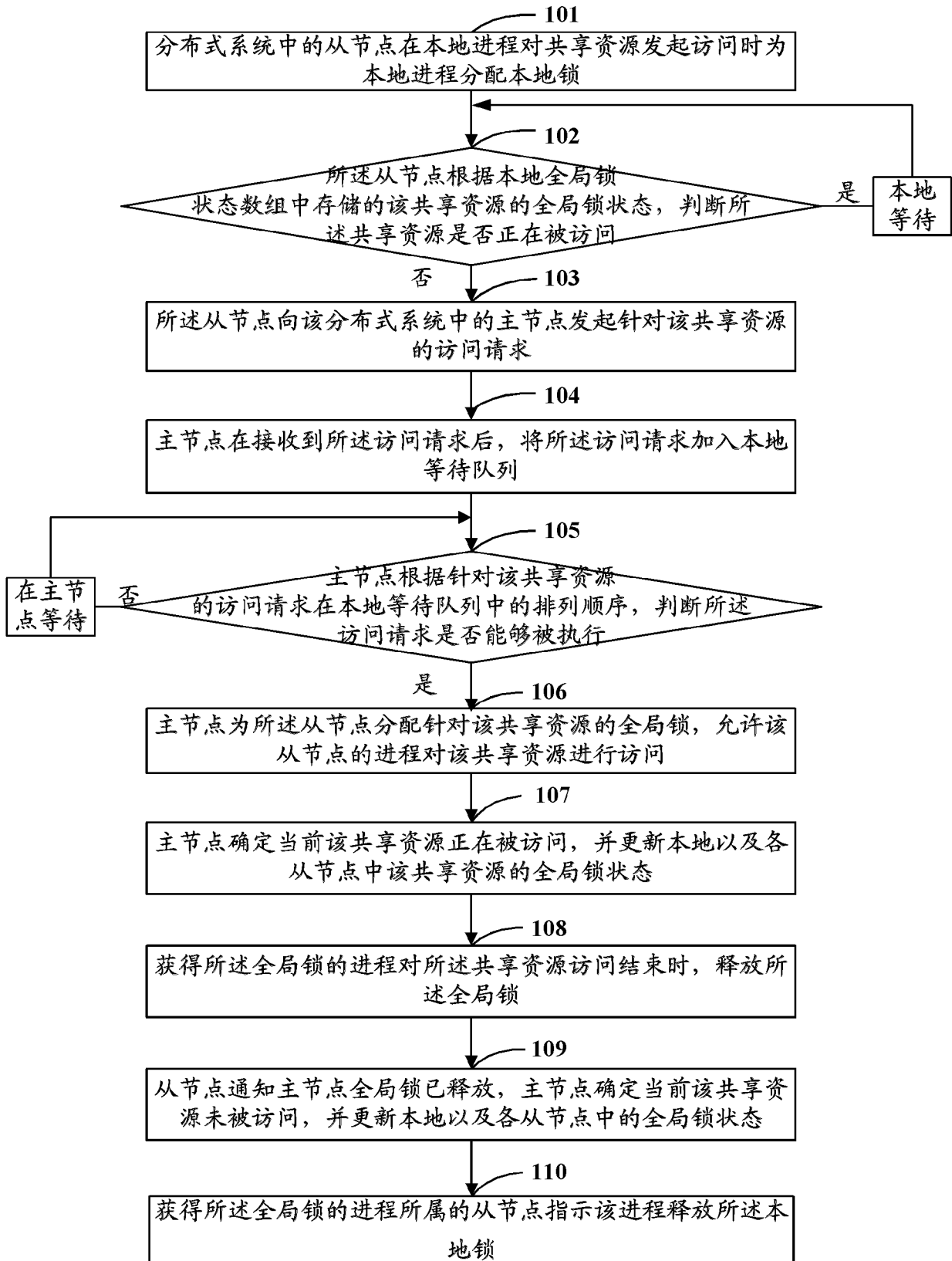


图 2

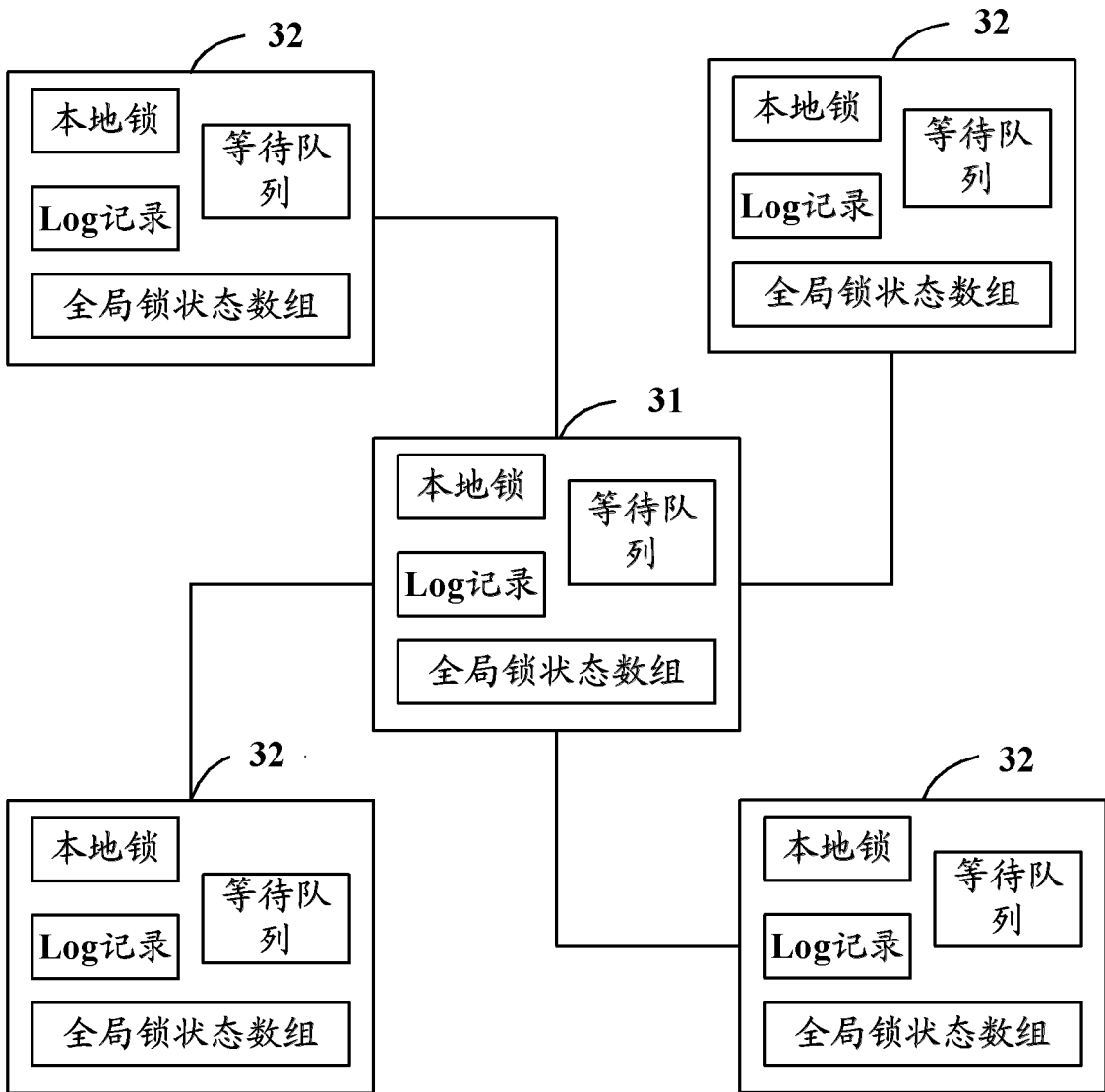


图 3