(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 117155930 B
(45)授权公告日 2024.02.06

(21)申请号 202311440542.6
(22)申请日 2023.11.01
(65)同一申请的已公布的文献号
申请公布号 CN 117155930 A
(43)申请公布日 2023.12.01
(73)专利权人 腾讯科技(深圳)有限公司
地址 518000 广东省深圳市南山区高新区科技中一路腾讯大厦35层
(72)发明人 杨一迪 黄辉
(74)专利代理机构 北京市立方律师事务所 11330
专利代理师 高梦露
(51)Int.Cl.
H04L 67/10(2022.01)
H04L 43/10(2022.01)

(56)对比文件
CN 116346588 A,2023.06.27
CN 116860421 A,2023.10.10
US 2019052520 A1,2019.02.14
US 7664125 B1,2010.02.16
CN 116055563 A,2023.05.02
CN 116225655 A,2023.06.06
WO 2023147750 A1,2023.08.10
CN 113886129 A,2022.01.04
CN 108134712 A,2018.06.08

审查员 周萍

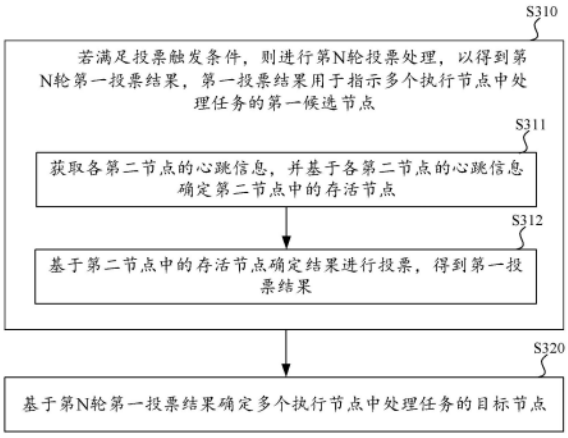
权利要求书3页 说明书20页 附图5页

(54)发明名称

分布式系统的节点确定方法、任务处理方法及相关装置

(57)摘要

本申请实施例提供了一种分布式系统的节点确定方法、任务处理方法及相关装置,涉及分布式技术领域,还可以涉及云技术领域。该方法包括:若满足投票触发条件,则进行第N轮投票处理,以得到第N轮第一投票结果,然后,基于第N轮第一投票结果确定多个执行节点中处理任务的目标节点,而每一轮投票处理,包括:获取各第二节点的心跳信息,并基于各第二节点的心跳信息确定第二节点中的存活节点,并且基于第二节点中的存活节点确定结果进行投票,得到第一投票结果,可以解决分布式技术中容易出现脑裂的技术问题,从而达到减少脑裂出现的情况的技术效果。



CN 117155930 B

1.一种分布式系统的节点确定方法,其特征在于,所述分布式系统包括多个执行节点,所述方法应用于所述多个执行节点中的任一第一节点,所述方法包括:

若满足投票触发条件,则进行第N轮投票处理,以得到第N轮第一投票结果,第一投票结果用于指示所述多个执行节点中处理任务的第一候选节点,N为不小于1的整数;

基于所述第N轮第一投票结果确定所述多个执行节点中处理任务的目标节点,其中,所述目标节点包括所述第一候选节点中的至少一个;

若所述目标节点包括所述第一节点,则确定所述第一节点处于工作状态;

若所述第一节点的最近一次心跳的时间与当前时间之间的时间间隔大于或等于时间间隔阈值,或当前满足投票触发条件且未确定最新的目标节点,则确定所述第一节点处于暂停状态;

其中,处于暂停状态或处于工作状态的第一节点均具有发起投票的能力;

其中,每一轮投票处理,包括:

获取各第二节点的心跳信息,并基于各第二节点的心跳信息确定第二节点中的存活节点,所述第二节点为所述多个执行节点中除所述第一节点外的节点,所述存活节点为最近一次心跳的时间与当前时间之间的时间间隔小于时间间隔阈值的节点;

基于所述第二节点中的存活节点确定结果进行投票,得到第一投票结果。

2.根据权利要求1所述的方法,其特征在于,所述投票触发条件包括以下的至少一项:

到达投票触发时间;

所述第一节点的最近一次心跳的时间与当前时间之间的时间间隔小于时间间隔阈值;

所述多个执行节点中的存活节点与第N-1轮投票所确定的目标节点不一致,N为不小于2的整数;

检测到所述第二节点中的存活节点发生变化。

3.根据权利要求1所述的方法,其特征在于,所述第一节点处于工作状态或暂停状态,处于工作状态的所述第一节点允许处理任务,处于暂停状态的所述第一节点禁止处理任务。

4.根据权利要求1所述的方法,其特征在于,所述基于所述第N轮第一投票结果确定所述多个执行节点中处理任务的目标节点,包括:

查询至少一个第二节点分别对应的第N轮第二投票结果,第二投票结果用于指示所述多个执行节点中处理任务的第二候选节点;

若所述第N轮第二投票结果查询失败,则将所述第N轮第一投票结果所指示的第一候选节点确定为目标节点;

若所述第N轮第二投票结果查询成功,则基于所述第N轮第一投票结果和所述第N轮第二投票结果确定所述多个执行节点中处理任务的目标节点。

5.根据权利要求4所述的方法,其特征在于,所述基于所述第N轮第一投票结果和所述第N轮第二投票结果确定所述多个执行节点中处理任务的目标节点,包括:

若基于所述第N轮第一投票结果和所述第N轮第二投票结果确定满足投票结束条件,则确定用于处理任务的目标节点集合,并将所述目标节点集合中的任一节点确定为目标节点,所述目标节点集合为所述第N轮第一投票结果所指示的第一候选节点和所述第N轮第二投票结果所指示的第二候选节点之间的交集或并集;

其中,所述投票结束条件包括以下的至少一项:

所述第N轮第一投票结果与各第二节点对应的第N轮第二投票结果一致;

查询到的第N轮第二投票结果所对应的所述至少一个第二节点与所述第二节点中的存活节点一致;

所述方法还包括:

若基于所述第N轮第一投票结果和所述第N轮第二投票结果确定不满足所述投票结束条件,则进行第N+1轮投票处理。

6.根据权利要求4所述的方法,其特征在于,所述多个执行节点中的任一节点与投票管理数据库连接,所述投票管理数据库用于记录各执行节点每一轮投票处理得到的投票结果;

所述查询至少一个第二节点分别对应的第N轮第二投票结果,包括:

从所述投票管理数据库中查询至少一个第二节点分别对应的第N轮第二投票结果。

7.根据权利要求6所述的方法,其特征在于,所述投票管理数据库还用于记录各执行节点上报的心跳信息;

所述获取各第二节点的心跳信息,包括:

从投票管理数据库中获取各第二节点的心跳信息。

8.根据权利要求1-7任一项所述的方法,其特征在于,所述基于所述第二节点中的存活节点确定结果进行投票,得到第一投票结果,包括以下的至少一项:

若所述存活节点确定结果为所述第二节点中的存活节点确定成功,则所述第一投票结果所指示的第一候选节点包括所述第一节点和所述第二节点中的存活节点;

若所述存活节点确定结果为所述第二节点中的存活节点确定失败,则所述第一投票结果所指示的第一候选节点包括所述第一节点且不包括所述第二节点中的存活节点。

9.一种分布式系统的任务处理方法,其特征在于,所述分布式系统包括多个执行节点,所述方法应用于所述多个执行节点中的任一第一节点,所述方法包括:

确定所述多个执行节点中用于处理任务的目标节点,其中,所述目标节点是基于如权利要求1-8任一项所述的方法确定的;

若所述目标节点包括所述第一节点,则确定待处理任务中的第一数量个目标任务,并对所述第一数量个目标任务进行处理。

10.根据权利要求9所述的方法,其特征在于,若所述目标节点为至少两个,则所述确定待处理任务中的第一数量个目标任务,包括以下的任一项:

确定所述目标节点的第二数量,并基于待处理任务的任务总数量和所述第二数量确定所述第一节点处理的第一数量个任务,其中,所述第一数量与所述第二数量负相关,所述第一数量与所述任务总数量正相关;

获取各目标节点的处理能力指示信息,并基于各处理能力指示信息所指示的任务处理能力确定各目标节点对应的比值系数,以及基于待处理任务的任务总数量和所述第一节点对应的比值系数确定第一节点处理的第一数量个任务,其中,各目标节点对应的比值系数与各目标节点对应的任务处理能力正相关,所述第一数量分别与所述第一节点对应的比值系数和所述任务总数量正相关。

11.一种分布式系统的节点确定装置,其特征在于,所述分布式系统包括多个执行节

点,所述装置应用于所述多个执行节点中的任一第一节点,所述装置包括:

决策模块,用于若满足投票触发条件,则进行第N轮投票处理,以得到第N轮第一投票结果,第一投票结果用于指示所述多个执行节点中处理任务的第一候选节点,N为不小于1的整数;

所述决策模块还用于基于所述第N轮第一投票结果确定所述多个执行节点中处理任务的目标节点,其中,所述目标节点包括所述第一候选节点中的至少一个;

所述决策模块还用于若所述目标节点包括所述第一节点,则确定所述第一节点处于工作状态;若所述第一节点的最近一次心跳的时间与当前时间之间的时间间隔大于或等于时间间隔阈值,或当前满足投票触发条件且未确定最新的目标节点,则确定所述第一节点处于暂停状态;其中,处于暂停状态或处于工作状态的第一节点均具有发起投票的能力;

其中,所述决策模块在进行每一轮投票处理时,用于:

获取各第二节点的心跳信息,并基于各第二节点的心跳信息确定第二节点中的存活节点,所述第二节点为所述多个执行节点中除所述第一节点外的节点,所述存活节点为最近一次心跳的时间与当前时间之间的时间间隔小于时间间隔阈值的节点;

基于所述第二节点中的存活节点确定结果进行投票,得到第一投票结果。

12.一种分布式系统的任务处理装置,其特征在于,所述分布式系统包括多个执行节点,所述装置应用于所述多个执行节点中的任一第一节点,所述装置包括:

决策模块,用于确定所述多个执行节点中用于处理任务的目标节点,其中,所述目标节点是基于如权利要求1-8任一项所述的方法确定的;

执行模块,用于若所述目标节点包括所述第一节点,则确定待处理任务中的第一数量个目标任务,并对所述第一数量个目标任务进行处理。

13.一种执行节点,包括存储器、处理器及存储在存储器上的计算机程序,其特征在于,所述处理器执行所述计算机程序以实现权利要求1-8任一项所述方法或者实现权利要求9-10任一项所述的方法。

14.一种分布式系统,其特征在于,包括多个执行节点,其中,所述多个执行节点的任一节点作为第一节点执行权利要求1-8任一项所述方法或者执行权利要求9-10任一项所述的方法。

15.一种计算机可读存储介质,其上存储有计算机程序,其特征在于,所述计算机程序被处理器执行时实现权利要求1-8任一项所述方法或者实现权利要求9-10任一项所述的方法。

分布式系统的节点确定方法、任务处理方法及相关装置

技术领域

[0001] 本申请涉及分布式技术领域,具体而言,本申请涉及一种分布式系统的节点确定方法、任务处理方法及相关装置。

背景技术

[0002] 在高可用的分布式技术中,会存在互为备份的多个执行节点,多个执行节点在处于连接状态时,由其中的一个执行节点处理任务,另一个执行节点作为备份执行节点,彼此通过心跳链路达成主备协商。但是,当多个执行节点出现心跳链路故障时,多个执行节点中只可以有一个为存活状态,即只可以由其中的一个执行节点处理任务,若多个执行节点都为存活状态,则同时处理任务,而且二者无法进行数据的同步,就表示出现了脑裂的状况,这种情况会导致任务数据的混乱。

[0003] 因此,如何减少脑裂出现的情况已成为了重要研究方向之一。

发明内容

[0004] 本申请实施例提供了一种分布式系统的节点确定方法、任务处理方法及相关装置,用于解决分布式技术中容易出现脑裂的技术问题,从而达到减少脑裂出现的情况的技术效果。

[0005] 一方面,本申请实施例提供了一种分布式系统的节点确定方法,分布式系统包括多个执行节点,方法应用于多个执行节点中的任一第一节点,方法包括:

[0006] 若满足投票触发条件,则进行第N轮投票处理,以得到第N轮第一投票结果,第一投票结果用于指示多个执行节点中处理任务的第一候选节点,N为不小于1的整数;

[0007] 基于第N轮第一投票结果确定多个执行节点中处理任务的目标节点,其中,目标节点包括第一候选节点中的至少一个;

[0008] 其中,每一轮投票处理,包括:

[0009] 获取各第二节点的心跳信息,并基于各第二节点的心跳信息确定第二节点中的存活节点,第二节点为多个执行节点中除第一节点外的节点,存活节点为最近一次心跳的时间与当前时间之间的时间间隔小于时间间隔阈值的节点;

[0010] 基于第二节点中的存活节点确定结果进行投票,得到第一投票结果。

[0011] 另一方面,本申请实施例还提供了一种分布式系统的任务处理方法,分布式系统包括多个执行节点,方法应用于多个执行节点中的任一第一节点,方法包括:

[0012] 确定多个执行节点中用于处理任务的目标节点,其中,目标节点是基于本申请任一实施例的方法确定的;

[0013] 若目标节点包括第一节点,则确定待处理任务中的第一数量个目标任务,并对第一数量个目标任务进行处理。

[0014] 另一方面,本申请实施例还提供了一种分布式系统的节点确定装置,分布式系统包括多个执行节点,装置应用于多个执行节点中的任一第一节点,装置包括:

[0015] 决策模块,用于若满足投票触发条件,则进行第N轮投票处理,以得到第N轮第一投票结果,第一投票结果用于指示多个执行节点中处理任务的第一候选节点,N为不小于1的整数;

[0016] 决策模块还用于基于第N轮第一投票结果确定多个执行节点中处理任务的目标节点,其中,目标节点包括第一候选节点中的至少一个;

[0017] 其中,决策模块在进行每一轮投票处理时,用于:

[0018] 获取各第二节点的心跳信息,并基于各第二节点的心跳信息确定第二节点中的存活节点,第二节点为多个执行节点中除第一节点外的节点,存活节点为最近一次心跳的时间与当前时间之间的时间间隔小于时间间隔阈值的节点;

[0019] 基于第二节点中的存活节点确定结果进行投票,得到第一投票结果。

[0020] 可选的,投票触发条件包括以下的至少一项:

[0021] 到达投票触发时间;

[0022] 第一节点的最近一次心跳的时间与当前时间之间的时间间隔小于时间间隔阈值;

[0023] 多个执行节点中的存活节点与第N-1轮投票所确定的目标节点不一致,N为不小于2的整数;

[0024] 检测到第二节点中的存活节点发生变化。

[0025] 可选的,第一节点处于工作状态或暂停状态,处于工作状态的第一节点允许处理任务,处于暂停状态的第一节点禁止处理任务;决策模块还用于:

[0026] 若确定出的目标节点包括第一节点,则将第一节点的状态设置为工作状态;

[0027] 若第一节点的最近一次心跳的时间与当前时间之间的时间间隔大于或等于时间间隔阈值,则将第一节点的状态设置为暂停状态;

[0028] 若满足投票触发条件且在确定出目标节点之前,将第一节点的状态设置为暂停状态。

[0029] 可选的,决策模块在基于第N轮第一投票结果确定多个执行节点中处理任务的目标节点时,可以用于:

[0030] 查询至少一个第二节点分别对应的第N轮第二投票结果,第二投票结果用于指示多个执行节点中处理任务的第二候选节点;

[0031] 若第N轮第二投票结果查询失败,则将第N轮第一投票结果所指示的第一候选节点确定为目标节点;

[0032] 若第N轮第二投票结果查询成功,则基于第N轮第一投票结果和第N轮第二投票结果确定多个执行节点中处理任务的目标节点。

[0033] 可选的,基于第N轮第一投票结果和第N轮第二投票结果确定多个执行节点中处理任务的目标节点时,可以用于:

[0034] 若基于第N轮第一投票结果和第N轮第二投票结果确定满足投票结束条件,则确定用于处理任务的目标节点集合,并将目标节点集合中的任一节点确定为目标节点,目标节点集合为第N轮第一投票结果所指示的第一候选节点和第N轮第二投票结果所指示的第二候选节点之间的交集或并集;

[0035] 其中,投票结束条件包括以下的至少一项:

[0036] 第N轮第一投票结果与各第二节点对应的第N轮第二投票结果一致;

[0037] 查询到的第N轮第二投票结果所对应的至少一个第二节点与第二节点中的存活节点一致；

[0038] 方法还包括：

[0039] 若基于第N轮第一投票结果和第N轮第二投票结果确定不满足投票结束条件，则进行第N+1轮投票处理。

[0040] 可选的，多个执行节点中的任一节点与投票管理数据库连接，投票管理数据库用于记录各执行节点每一轮投票处理得到的投票结果；

[0041] 决策模块在查询至少一个第二节点分别对应的第N轮第二投票结果时，可以用于：

[0042] 从投票管理数据库中查询至少一个第二节点分别对应的第N轮第二投票结果。

[0043] 可选的，投票管理数据库还用于记录各执行节点上报的心跳信息；决策模块在获取各第二节点的心跳信息时，可以用于：

[0044] 从投票管理数据库中获取各第二节点的心跳信息。

[0045] 可选的，决策模块在基于第二节点中的存活节点确定结果进行投票，得到第一投票结果时，可以用于以下的至少一项：

[0046] 若存活节点确定结果为第二节点中的存活节点确定成功，则第一投票结果所指示的第一候选节点包括第一节点和第二节点中的存活节点；

[0047] 若存活节点确定结果为第二节点中的存活节点确定失败，则第一投票结果所指示的第一候选节点包括第一节点且不包括第二节点中的存活节点。

[0048] 另一方面，本申请实施例还提供了一种分布式系统的任务处理装置，分布式系统包括多个执行节点，装置应用于多个执行节点中的任一第一节点，装置包括：

[0049] 决策模块，用于确定多个执行节点中用于处理任务的目标节点，其中，目标节点是基于本申请任一实施例的方法确定的；

[0050] 执行模块，用于若目标节点包括第一节点，则确定待处理任务中的第一数量个目标任务，并对第一数量个目标任务进行处理。

[0051] 可选的，若目标节点为至少两个，则执行模块在确定待处理任务中的第一数量个目标任务，可以用于以下的任一项：

[0052] 确定目标节点的第二数量，并基于待处理任务的任务总数量和第二数量确定第一节点处理的第一数量个任务，其中，第一数量与第二数量负相关，第一数量与任务总数量正相关；

[0053] 获取各目标节点的处理能力指示信息，并基于各处理能力指示信息所指示的任务处理能力确定各目标节点对应的比值系数，以及基于待处理任务的任务总数量和第一节点对应的比值系数确定第一节点处理的第一数量个任务，其中，各目标节点对应的比值系数与各目标节点对应的任务处理能力正相关，第一数量分别与第一节点对应的比值系数和任务总数量正相关。

[0054] 另一方面，本申请实施例还提供了一种执行节点，包括存储器、处理器及存储在存储器上的计算机程序，处理器执行计算机程序以实现本申请任一实施例的方法。

[0055] 另一方面，本申请实施例还提供了一种分布式系统，包括多个执行节点，其中，多个执行节点的任一节点作为第一节点执行本申请任一实施例的方法。

[0056] 另一方面，本申请实施例还提供了一种计算机可读存储介质，其上存储有计算机

程序,计算机程序被处理器执行时实现本申请任一实施例的方法。

[0057] 另一方面,本申请实施例还提供了一种计算机程序产品,包括计算机程序,计算机程序被处理器执行时实现本申请任一实施例的方法。

[0058] 本实施例的技术方案,通过若满足投票触发条件,则进行第N轮投票处理,以得到第N轮第一投票结果,第一投票结果用于指示多个执行节点中处理任务的第一候选节点,基于第N轮第一投票结果确定多个执行节点中处理任务的目标节点,而每一轮投票处理,包括:获取各第二节点的心跳信息,并基于各第二节点的心跳信息确定第二节点中的存活节点,第二节点为多个执行节点中除第一节点外的节点,存活节点为最近一次心跳的时间与当前时间之间的时间间隔小于时间间隔阈值的节点;基于第二节点中的存活节点确定结果进行投票,得到第一投票结果,也就是说,在每一轮投票时,可以基于当前存活的节点来确定出用于处理任务的目标节点,从而持续更新处理任务的目标节点,由此可以解决分布式技术中容易出现脑裂的技术问题,从而达到减少脑裂出现的情况的技术效果。

附图说明

[0059] 为了更清楚地说明本申请实施例中的技术方案,下面将对本申请实施例描述中所需要使用的附图作简单地介绍。

[0060] 图1为本申请实施例提供的一种方案实施的框架示意图;

[0061] 图2为本申请实施例提供的一种执行节点的模块架构示意图;

[0062] 图3为本申请实施例提供的一种分布式系统的节点确定方法的流程示意图;

[0063] 图4为本申请实施例提供的一种节点心跳更新的流程示意图;

[0064] 图5为本申请实施例提供的一种节点在不同状态下发起投票的示意图;

[0065] 图6为本申请实施例提供的一种节点的状态更新流程示意图;

[0066] 图7为本申请实施例提供的一种分布式系统的任务处理方法的流程示意图;

[0067] 图8为本申请实施例提供的一种两个执行节点的CPU使用率的对比示意图;

[0068] 图9为本申请实施例提供的一种分布式系统的节点确定装置的结构示意图;

[0069] 图10为本申请实施例提供的一种分布式系统的任务处理装置的结构示意图;

[0070] 图11为本申请实施例提供的一种执行节点的结构示意图。

具体实施方式

[0071] 下面结合本申请中的附图描述本申请的实施例。应理解,下面结合附图所阐述的实施方式,是用于解释本申请实施例的技术方案的示例性描述,对本申请实施例的技术方案不构成限制。

[0072] 本技术领域技术人员可以理解,除非特意声明,这里使用的单数形式“一”、“一个”、和“该”也可包括复数形式。应该进一步理解的是,本申请实施例所使用的术语“包括”以及“包含”是指相应特征可以实现为所呈现的特征、信息、数据、步骤、操作、元件和/或组件,但不排除实现为本技术领域所支持其他特征、信息、数据、步骤、操作、元件、组件和/或它们的组合等。应该理解,当我们称一个元件被“连接”或“耦接”到另一元件时,该一个元件可以直接连接或耦接到另一元件,也可以指该一个元件和另一元件通过中间元件建立连接关系。此外,这里使用的“连接”或 “耦接”可以包括无线连接或无线耦接。这里使用的术语

“和/或”指示该术语所限定的项目中的至少一个,例如“A和/或B”指示实现为“A”,或者实现为“A”,或者实现为“A和B”。“多个”可以是指至少两个。

[0073] 为使本申请的目的、技术方案和优点更加清楚,下面将结合附图对本申请实施方式作进一步地详细描述。

[0074] 首先对本申请涉及的几个名词进行介绍和解释:

[0075] 任务:计算机领域中根据业务场景需要执行的任务。

[0076] 执行节点:执行任务的单元,可以是某台计算机上的一个进程。

[0077] 容灾:应对故障的能力,如存在多个执行节点时,某个执行节点故障后(计算机掉电、网络故障等),任务能否被剩下的执行节点继续执行的能力。

[0078] 负载均衡:分摊执行任务的能力。如单位时间内各执行节点执行的任务数是否接近。

[0079] 脑裂:在任务管理场景中,当相同任务同时被多个执行节点执行时,可能导致任务在执行过程中相互影响,出现执行结果未知的情况。

[0080] 在相关技术中,对于分布式系统,主要的方案包括但不限于如下所示:

[0081] 方案一:基于分布式协议对执行节点进行选主,选主成功的执行节点可以执行任务,其他执行节点以灾备的角色处于空闲状态,当主节点异常时,其他执行节点重新选主并执行任务。

[0082] 方案二:引入消息中间件,以生产者-消费者模式,让多个执行节点消费任务消息,并执行任务。其中,生产者通过分布式协议进行选主,使多个生产者以主备的方式运行,选主成功的生产者将任务集发送到消息中间件后,多个消费者通过竞争的方式从消息中间件中获取任务信息并执行。

[0083] 方案三:基于负载检测进行任务调度,对于负载超过阈值的执行节点,任务调度模块会暂停将任务分配到该节点上,待负载恢复到阈值以内后再将任务分配到该节点;

[0084] 在相关技术中,都存在各自的缺陷:

[0085] 方案一的缺陷:

[0086] 该方案具备容灾能力,但不具备负载均衡的能力。所有的任务都在选主成功的执行节点上执行,会导致负载很高,甚至达到计算机资源瓶颈,使得任务无法及时完成。同时,其他执行节点处于空闲状态,造成了计算机资源的浪费。

[0087] 方案二的缺陷:

[0088] 1)该方案较为复杂,引入了消息中间件,使用者需要具备对消息中间件有一定维护能力;

[0089] 2)该方案存在着一些额外的资源消耗。消息中间件处理着任务在生产者-消费者之间的传递,需要消耗计算机资源,尤其是任务量巨大时,产生的资源消耗更加严重。

[0090] 方案三的缺陷:

[0091] 该方案不能解决相同任务的脑裂执行问题。任务执行超时时,任务的实际执行情况可能仍然在执行中,如果任务超时发起重试可能导致相同任务同时执行,出现脑裂的情况。

[0092] 针对相关技术中所存在的上述至少一个技术问题或需要改善的地方,本申请提出一种分布式系统的节点确定方法、任务处理方法及相关装置,在提供容灾及负载均衡能力

的前提下,保证相同任务不会被多个执行节点同时执行,避免出现脑裂的情况,引起任务执行时的相互影响。

[0093] 该方案通过若满足投票触发条件,则进行第N轮投票处理,以得到第N轮第一投票结果,第一投票结果用于指示多个执行节点中处理任务的第一候选节点,N为不小于1的整数,然后基于第N轮第一投票结果确定多个执行节点中处理任务的目标节点;而每一轮投票处理,包括:获取各第二节点的心跳信息,并基于各第二节点的心跳信息确定第二节点中的存活节点,第二节点为多个执行节点中除第一节点外的节点,存活节点为最近一次心跳的时间与当前时间之间的时间间隔小于时间间隔阈值的节点;基于第二节点中的存活节点确定结果进行投票,得到第一投票结果,可以解决分布式技术中容易出现脑裂的技术问题,从而达到减少脑裂出现的情况的技术效果。此外,与相关技术相比较,本方案可以防止任务并行执行,无论是组件异常导致的切换,还是任务执行超时,都会保证任务的串行执行,避免并行执行带来的相互影响。此外,执行节点之间负载均衡,实现横向扩容的能力。此外,本申请的技术方案为一种轻量级方案,所使用的算法可以静态库(LIB)的方式嵌入到各个业务模块中使用。

[0094] 云技术(Cloud technology)是指在广域网或局域网内将硬件、软件、网络等系列资源统一起来,实现数据的计算、储存、处理和共享的一种托管技术。

[0095] 云技术(Cloud technology)基于云计算商业模式应用的网络技术、信息技术、整合技术、管理平台技术、应用技术等的总称,可以组成资源池,按需所用,灵活便利。云计算技术将变成重要支撑。技术网络系统的后台服务需要大量的计算、存储资源,如视频网站、图片类网站和更多的门户网站。伴随着互联网行业的高度发展和应用,将来每个物品都有可能存在自己的识别标志,都需要传输到后台系统进行逻辑处理,不同程度级别的数据将会分开处理,各类行业数据皆需要强大的系统后盾支撑,只能通过云计算来实现。

[0096] 具体的,可以涉及云计算。例如,通过云计算进行投票得到投票结果。云计算(cloud computing)是一种计算模式,它将计算任务分布在大量计算机构成的资源池上,使各种应用系统能够根据需要获取计算力、存储空间和信息服务。提供资源的网络被称为“云”。“云”中的资源在使用者看来是可以无限扩展的,并且可以随时获取,按需使用,随时扩展,按使用付费。作为云计算的基础能力提供商,会建立云计算资源池(简称云平台,一般称为IaaS(Infrastructure as a Service,基础设施即服务)平台,在资源池中部署多种类型的虚拟资源,供外部客户选择使用。云计算资源池中主要包括:计算设备(为虚拟化机器,包含操作系统)、存储设备、网络设备。按照逻辑功能划分,在IaaS(Infrastructure as a Service,基础设施即服务)层上可以部署PaaS(Platform as a Service,平台即服务)层,PaaS层之上再部署SaaS (Software as a Service,软件即服务)层,也可以直接将SaaS部署在IaaS上。PaaS为软件运行的平台,如数据库、web容器等。SaaS为各式各样的业务软件,如web门户网站、短信群发器等。一般来说,SaaS和PaaS相对于IaaS是上层。

[0097] 具体的,还可涉及云存储,例如通过云存储方案实施所需要的信息,例如心跳信息,投票结果等。

[0098] 云存储(cloud storage)是在云计算概念上延伸和发展出来的一个新的概念,分布式云存储系统 (以下简称存储系统)是指通过集群应用、网格技术以及分布存储文件系统等功能,将网络中大量各种不同类型的存储设备(存储设备也称之为存储节点)通过应用

软件或应用接口集合起来协同工作,共同对外提供数据存储和业务访问功能的一个存储系统。

[0099] 目前,存储系统的存储方法为:创建逻辑卷,在创建逻辑卷时,就为每个逻辑卷分配物理存储空间,该物理存储空间可能是某个存储设备或者某几个存储设备的磁盘组成。客户端在某一逻辑卷上存储数据,也就是将数据存储在文件系统上,文件系统将数据分成许多部分,每一部分是一个对象,对象不仅包含数据而且还包含数据标识(ID,ID entity)等额外的信息,文件系统将每个对象分别写入该逻辑卷的物理存储空间,且文件系统会记录每个对象的存储位置信息,从而当客户端请求访问数据时,文件系统能够根据每个对象的存储位置信息让客户端对数据进行访问。

[0100] 存储系统为逻辑卷分配物理存储空间的过程,具体为:按照对存储于逻辑卷的对象的容量估量(该估量往往相对于实际要存储的对象的容量有很大余量)和独立冗余磁盘阵列(RAID,Redundant Array of Independent Disk)的组别,预先将物理存储空间划分成分条,一个逻辑卷可以理解为一个分条,从而为逻辑卷分配了物理存储空间。

[0101] 下面通过对几个示例性实施方式的描述,对本申请实施例的技术方案以及本申请的技术方案产生的技术效果进行说明。需要指出的是,下述实施方式之间可以相互参考、借鉴或结合,对于不同实施方式中相同的术语、相似的特征以及相似的实施步骤等,不再重复描述。

[0102] 请参阅图1,图1为本申请实施例提供的一种方案实施的框架示意图。如图1所示的方案实施框架可以包括多个执行节点110(以下也简称为节点)。

[0103] 其中,各执行节点110用于处理任务(也称执行任务)。以应用场景为打车为例,则任务可以为派单任务,则各执行节点110可以用于处理派单任务。

[0104] 在本实施例中,可以将多个执行节点110中的任一节点作为第一节点,从而执行本申请任一实施例的方法的步骤,也即在本方案的实施过程中,可以是各执行节点110均执行本申请任一实施例的方法的步骤,从而实现分布式系统的节点确定或者实现分布式系统的任务处理中的至少一项。执行节点110可以是具备数据处理能力的服务器或者服务器集群。

[0105] 可选的,多个执行节点110之间可以进行通信。

[0106] 可选的,方案实施框架还可以包括投票管理数据库120,各执行节点110可以与投票管理数据库120之间进行通信,则各节点可以将方案实施所需要的信息存储至投票管理数据库120中,以供其他节点从投票管理数据库120中获取需要的信息,多个执行节点110之间可以不需要通信。其中,投票管理数据库120记录有任务管理元数据。任务管理元数据用于记录任务管理所需的节点心跳信息、投票结果和以及任务分工的决策信息。该元数据需要具备高可用和一致性保证,可使用基于分布式协议的元数据存储,或关系型数据库(关系型数据库的故障切换能力需要额外的管理系统负责)。

[0107] 可选的,方案实施框架还可以包括业务数据库130。其中,业务数据库130记录有业务元数据。任务的执行通常会依赖业务相关的元数据,如任务可能来源于业务元数据。业务元数据与任务管理元数据无关,可以是不同的存储方案。

[0108] 需要说明的是,数据库可以是关系型数据库,数据库中的元数据可以是包括执行节点110心跳元数据和决策元数据。心跳元数据和决策元数据可以是关系型数据库的数据形式。

[0109] 可以理解的是,可以根据需要设置多个节点之间的通信方式,在此不做限定。

[0110] 请参阅图2,图2为本申请实施例提供的一种执行节点的模块架构示意图。

[0111] 任务管理相关模块以LIB的形式嵌入到普通的任务执行节点中,提供决策出本执行节点需要执行的任务分片的能力,执行模块根据决策结果执行相应任务分片即可。接下来对各模块组件进行简单介绍:

[0112] 任务管理元数据:记录任务管理所需的节点心跳信息,以及任务分工的决策信息。该元数据需要具备高可用和一致性保证,可使用基于分布式协议的元数据存储,或关系型数据库(关系型数据库的故障切换能力需要额外的管理系统负责)。后续介绍中会使用关系型数据库进行说明。

[0113] 上报模块:负责定时上报任务执行节点的心跳信息或上报投票结果中的至少一项。

[0114] 决策模块:根据各任务执行节点的心跳信息,决策出存活的执行节点,以及各节点应该执行的任务分片信息,包括总分片数、本节点执行的分片序号。

[0115] 执行模块:根据总分片数、本节点分片序号,获取分片任务集并处理任务。

[0116] 业务元数据:任务的处理通常会依赖业务相关的元数据,如任务可能来源于业务元数据。业务元数据与任务管理元数据无关,可以是不同的存储方案。

[0117] 请参阅图3,图3为本申请实施例提供的一种分布式系统的节点确定方法的流程示意图。本实施例的方法可以由分布式系统的任一节点执行。为了便于区分,将执行方法的节点作为第一节点,其他节点作为第二节点,而由于可以是各节点均执行本申请实施例的方法,因此,各节点即可以是第一节点,也可以是第二节点,根据与自身的相对关系来确定其是第一节点还是第二节点,在此不做限定。

[0118] 如图3所示的分布式系统的节点确定方法可以包括:

[0119] S310、若满足投票触发条件,则进行第N轮投票处理,以得到第N轮第一投票结果,第一投票结果用于指示多个执行节点中处理任务的第一候选节点。

[0120] 其中,N为不小于1的整数。投票触发条件可以是指触发第一节点进行投票的条件,投票的目的在于确定出多个执行节点中用于处理任务的目标节点。可选的,投票触发条件可以是在确定第一节点可以作为目标节点此因素,并且结合其他因素所确定出的条件,则第一候选节点可以包括第一节点,本实施例对于具体的投票触发条件不做限定。

[0121] 在本实施例中,可以是每一轮投票处理,则得到一轮第一投票结果,则第N轮投票处理对应得到的是第N轮第一投票结果。

[0122] 其中,每一轮投票处理,可以包括:

[0123] S311、获取各第二节点的心跳信息,并基于各第二节点的心跳信息确定第二节点中的存活节点。

[0124] 其中,第二节点为多个执行节点中除第一节点外的节点。存活节点为最近一次心跳的时间与当前时间之间的时间间隔小于时间间隔阈值的节点。心跳信息可以包括上报的各心跳对应的心跳时间。在本实施例中,具体的,可以是各节点上报心跳,则接收上报的心跳的主体可以记录节点上报心跳的时间或记录主体接收到心跳的时间,作为心跳对应的心跳时间,则节点每上报一次心跳,则可以记录一次心跳时间,由此可以形成各节点对应的心跳信息。

[0125] 需要说明的是,可以是任意两个节点之间进行通信,则第一节点可以获取各第二节点的心跳信息;此外,还可以是各节点与投票管理数据库之间通信,从而向投票管理数据库上报心跳,则投票管理数据库可以存储各节点的心跳信息。

[0126] 可选的,各节点的心跳信息可以存储在同一列表中,也可以存储在不同列表中,在此不做限定。

[0127] 请参阅表1,表1为本申请实施例提供的一种存储节点的心跳信息的示意。

[0128] 表1

[0129]

节点标识（Worker ip）	心跳时间
9.0.0.1	2021-09-10 17:00:03
9.0.0.2	2021-09-10 17:00:02
9.0.0.3	2021-09-10 16:59:50
9.0.0.4	2021-09-10 17:00:01

[0130] 请参阅图4,图4为本申请实施例提供的一种节点心跳更新的流程示意图。

[0131] 如图4所示,每隔n秒定时更新心跳,更新成功后,延长任务的处理截止时间。其中,任务处理截止时间=心跳更新时间+配置的处理超时时间。

[0132] S312、基于第二节点中的存活节点确定结果进行投票,得到第一投票结果。

[0133] 在本实施例中,可选的,基于第二节点中的存活节点确定结果进行投票,可以是将多个节点中当前存活的节点,确定为第一投票结果所指示的第一候选节点。

[0134] 在一种可能的实现方式中,基于第二节点中的存活节点确定结果进行投票,得到第一投票结果,包括以下的至少一项:

[0135] 若存活节点确定结果为第二节点中的存活节点确定成功,则第一投票结果所指示的第一候选节点包括第一节点和第二节点中的存活节点;

[0136] 若存活节点确定结果为第二节点中的存活节点确定失败,则第一投票结果所指示的第一候选节点包括第一节点且不包括第二节点中的存活节点。

[0137] 在本实施例中,具体的,若存活节点确定结果为第二节点中的存活节点确定成功,则说明第二节点中有存活节点,则此时可以将第一节点和第二节点中的存活节点作为第一候选节点。若存活节点确定结果为第二节点中的存活节点确定失败,则说明第二节点中无存活节点,则此时可以将第一节点作为第一候选节点。

[0138] S320、基于第N轮第一投票结果确定多个执行节点中处理任务的目标节点。

[0139] 其中,目标节点可以包括第一候选节点中的至少一个。在本实施例中,确定出目标节点后,可以通过目标节点处理待处理任务。

[0140] 本实施例的技术方案,通过若满足投票触发条件,则进行第N轮投票处理,以得到第N轮第一投票结果,第一投票结果用于指示多个执行节点中处理任务的第一候选节点,基于第N轮第一投票结果确定多个执行节点中处理任务的目标节点,而每一轮投票处理,包括:获取各第二节点的心跳信息,并基于各第二节点的心跳信息确定第二节点中的存活节点,第二节点为多个执行节点中除第一节点外的节点,存活节点为最近一次心跳的时间与当前时间之间的时间间隔小于时间间隔阈值的节点;基于第二节点中的存活节点确定结果

进行投票,得到第一投票结果,也就是说,在每一轮投票时,可以基于当前存活的节点来确定出用于处理任务的目标节点,从而持续更新处理任务的目标节点,由此可以解决分布式技术中容易出现脑裂的技术问题,从而达到减少脑裂出现的情况的技术效果。

[0141] 在一种可能的实现方式中,投票触发条件包括以下的至少一项:

[0142] 到达投票触发时间;

[0143] 第一节点的最近一次心跳的时间与当前时间之间的时间间隔小于时间间隔阈值;

[0144] 多个执行节点中的存活节点与第N-1轮投票所确定的目标节点不一致,N为不小于2的整数;

[0145] 检测到第二节点中的存活节点发生变化。

[0146] 对于投票触发条件包括投票触发时间来说,投票触发时间可以是持续变化的,可以预先设定好投票触发间隔,则下一次投票触发时间可以是上一次投票触发时间加上投票触发间隔。通过到达投票触发时间来触发投票,可以定时触发投票。需要说明的是,可以是各节点之间,第一次投票触发的时间相同。

[0147] 对于投票触发条件包括第一节点的最近一次心跳的时间与当前时间之间的时间间隔小于时间间隔阈值来说,此时可以确保第一节点为存活节点,也即第一节点比较稳定,能够进行节点选举或者是能够处理任务。具体的,本实施例的第一节点可以处于工作状态或暂停状态,处于工作状态的第一节点允许处理任务,处于暂停状态的第一节点禁止处理任务。对于处于工作状态的第一节点来说,若第一节点的最近一次心跳的时间与当前时间之间的时间间隔小于时间间隔阈值,则说明第一节点一直在线,则可以进行投票。对于处于暂停状态的第一节点来说,若第一节点的最近一次心跳的时间与当前时间之间的时间间隔小于时间间隔阈值,则说明第一节点恢复在线,则可以进行投票。

[0148] 对于投票触发条件包括多个执行节点中的存活节点与第N-1轮投票所确定的目标节点不一致来说,此时说明最新的存活节点与最近一次投票所确定的目标节点不一致,此时需要重新确定目标节点,否则具有脑裂的风险。具体的,多个执行节点中的存活节点与第N-1轮投票所确定的目标节点不一致,可以是多个执行节点中的存活节点少于或多余第N-1轮投票所确定的目标节点,当多个执行节点中的存活节点与第N-1轮投票所确定的目标节点一一对应时,认为多个执行节点中的存活节点与第N-1轮投票所确定的目标节点一致。

[0149] 对于投票触发条件包括检测到第二节点中的存活节点发生变化来说,此时说明最新的存活节点与最近一次投票所确定的目标节点不一致,此时需要重新确定目标节点,否则具有脑裂的风险。

[0150] 需要说明的是,可以根据需要选择以上的一种或多种条件作为投票触发条件,在此不做限定。

[0151] 可以理解的是,通过选择以上的多种条件作为投票触发条件,可以从多个角度确定触发投票的时机,从而提高投票触发的准确性以及节约投票所需要的运算资源。

[0152] 在一种可能的实现方式中,第一节点处于工作状态或暂停状态,处于工作状态的第一节点允许处理任务,处于暂停状态的第一节点禁止处理任务;方法还包括以下的至少一项:

[0153] 若确定出的目标节点包括第一节点,则将第一节点的状态设置为工作状态;

[0154] 若第一节点的最近一次心跳的时间与当前时间之间的时间间隔大于或等于时间

间隔阈值,则将第一节点的状态设置为暂停状态;

[0155] 若满足投票触发条件且在确定出目标节点之前,将第一节点的状态设置为暂停状态。

[0156] 具体的,若确定出的目标节点包括第一节点,则说明第一节点可以处理任务,则将第一节点的状态设置为工作状态。若第一节点的最近一次心跳的时间与当前时间之间的时间间隔大于或等于时间间隔阈值,则说明第一节点运行不稳定,若继续工作有较大风险发生脑裂问题,因此,将第一节点的状态设置为暂停状态,以禁止其继续处理任务。若满足投票触发条件且在确定出目标节点之前,则说明已经开始投票但最新的目标节点还未确定,此时若继续工作有较大风险发生脑裂问题,因此将第一节点的状态设置为暂停状态。

[0157] 需要说明的是,将第一节点的状态设置为工作状态,可以是维持第一节点的工作状态,也可以是将第一节点从暂停状态切换至工作状态,在此不做限定。同理,将第一节点的状态设置为暂停状态,可以是维持第一节点的暂停状态,也可以是将第一节点从工作状态切换至暂停状态,在此不做限定。

[0158] 在本实施例中,处于暂停状态或处于工作状态的第一节点均可以发起投票。如图5所示,图5为本申请实施例提供的一种节点在不同状态下发起投票的示意图。

[0159] 如图5所示,只有在工作状态(running)时,才可以获取任务并执行;心跳超时时进入暂停状态(pause),不再执行任务;进入暂停状态后会尝试发起投票(voting),当确定出的目标节点包括第一节点时,也即本节点可以属于参与工作的节点时,重新进入工作状态。

[0160] 本实施例的技术方案,通过根据不同时机情况设定第一节点处于工作状态或暂停状态,可以进一步减少脑裂问题发生的概率,进一步减少脑裂出现的情况。

[0161] 以下实施例在以上任一实施例的基础上,对于如何基于第N轮第一投票结果确定多个执行节点中处理任务的目标节点进行进一步说明。

[0162] 在一种可能的实现方式中,基于第N轮第一投票结果确定多个执行节点中处理任务的目标节点,可以包括:

[0163] 将第N轮第一投票结果所指示的第一候选节点确定为目标节点。

[0164] 在另一种可能的实现方式中,基于第N轮第一投票结果确定多个执行节点中处理任务的目标节点,包括:

[0165] 查询至少一个第二节点分别对应的第N轮第二投票结果,第二投票结果用于指示多个执行节点中处理任务的第二候选节点;

[0166] 若第N轮第二投票结果查询失败,则将第N轮第一投票结果所指示的第一候选节点确定为目标节点;

[0167] 若第N轮第二投票结果查询成功,则基于第N轮第一投票结果和第N轮第二投票结果确定多个执行节点中处理任务的目标节点。

[0168] 在本实施例中,具体的,各节点可以根据自身的实际情况选择是否触发投票,若至少一个第二节点也触发投票,则可以获取到至少一个第二节点分别对应的第N轮第二投票结果,若没有第二节点触发投票,则无法获取到第二节点对应的第N轮第二投票结果。

[0169] 本实施例的技术方案,通过在第N轮第二投票结果查询失败的情况下才将第N轮第一投票结果所指示的第一候选节点确定为目标节点,若第N轮第二投票结果查询成功,则基于第N轮第一投票结果和第N轮第二投票结果确定多个执行节点中处理任务的目标节点,可

以提高目标节点的确定准确性,进一步减少脑裂的发生概率。

[0170] 在一种可能的实现方式中,基于第N轮第一投票结果和第N轮第二投票结果确定多个执行节点中处理任务的目标节点,可以包括:

[0171] 确定用于处理任务的目标节点集合,并将目标节点集合中的任一节点确定为目标节点,目标节点集合为第N轮第一投票结果所指示的第一候选节点和第N轮第二投票结果所指示的第二候选节点之间的交集或并集。

[0172] 在另一种可能的实现方式中,基于第N轮第一投票结果和第N轮第二投票结果确定多个执行节点中处理任务的目标节点,包括:

[0173] 若基于第N轮第一投票结果和第N轮第二投票结果确定满足投票结束条件,则确定用于处理任务的目标节点集合,并将目标节点集合中的任一节点确定为目标节点,目标节点集合为第N轮第一投票结果所指示的第一候选节点和第N轮第二投票结果所指示的第二候选节点之间的交集或并集。

[0174] 在本实施例中,选择交集还是并集可以根据需要设置。具体的,选择交集得到的目标节点集合中的节点的数量,小于或等于选择并集得到的目标节点集合中的节点的数量。

[0175] 可以理解的是,选择交集作为目标节点集合,可以进一步减少脑裂情况;而选择并集作为目标节点集合,可以进一步提高处理任务的能力。

[0176] 可选的,若待处理任务的当前数量小于任务数量阈值,则可以选择交集作为目标节点结合,从而减少脑裂情况;若待处理任务的当前数量大于或等于任务数量阈值,则可以选择并集作为目标节点集合,可以进一步提高处理任务的能力。

[0177] 其中,投票结束条件包括以下的至少一项:

[0178] 第N轮第一投票结果与各第二节点对应的第N轮第二投票结果一致;

[0179] 查询到的第N轮第二投票结果所对应的至少一个第二节点与第二节点中的存活节点一致。

[0180] 在本实施例中,第N轮第一投票结果与各第二节点对应的第N轮第二投票结果一致,可以是第N轮第一投票结果所指示的第一候选节点,与个第二节点对应的第N轮第二投票结果所指示的第二候选节点一致。具体的,第N轮第一投票结果与各第二节点对应的第N轮第二投票结果一致,则交集或并集的结果相同,均为第N轮第一投票结果,则此时相当于将第N轮第一投票结果所指示的第一候选节点确定为目标节点。

[0181] 查询到的第N轮第二投票结果所对应的至少一个第二节点与第二节点中的存活节点一致,则说明存活的各节点均发起了投票。

[0182] 在本实施例中,通过第N轮第一投票结果与各第二节点对应的第N轮第二投票结果一致和/或查询到的第N轮第二投票结果所对应的至少一个第二节点与第二节点中的存活节点一致,才确定为目标节点,可以提高目标节点的确定准确性,进一步降低脑裂的情况。

[0183] 可选的,本实施例的方法还可以包括:

[0184] 若基于第N轮第一投票结果和第N轮第二投票结果确定不满足投票结束条件,则进行第N+1轮投票处理。

[0185] 在本实施例中,若基于第N轮第一投票结果和第N轮第二投票结果确定不满足投票结束条件,则说明此时尚未能够确定出目标节点,因此,需要进行第N+1轮投票处理,直至满足投票结束条件。

[0186] 为了便于理解,以下实施例提供一些确定出目标节点的例子进行辅助说明。

[0187] 请参阅表2,表2为本申请实施例提供的一种参与处理任务的目标节点的示意。

[0188] 表2

[0189]

ballot	voter	vote	vote Time
1	9.0.0.1	9.0.0.1	2021-09-10 17:00:03
2	9.0.0.2	9.0.0.1;9.0.0.2	2021-09-10 17:01:03
2	9.0.0.1	9.0.0.1;9.0.0.2	2021-09-10 17:01:13
3	9.0.0.3	9.0.0.1;9.0.0.2;9.0.0.3	2021-09-10 17:02:23
3	9.0.0.1	9.0.0.1;9.0.0.2;9.0.0.3	2021-09-10 17:02:23
3	9.0.0.2	9.0.0.1;9.0.0.2;9.0.0.3	2021-09-10 17:02:33
4	9.0.0.3	9.0.0.2;9.0.0.3	2021-09-10 17:03:03
4	9.0.0.2	9.0.0.2;9.0.0.3	2021-09-10 17:03:03

[0190] 其中,ballot为投票编号,表示第几轮投票;voter为投票参与者,本例中使用ip作为标识;vote为投票内容,是参与处理任务的目标节点列表。

[0191] 本例演示了这样一个决策过程:

[0192] a)9.0.0.1启动,触发第一轮投票并决策出由9.0.0.1执行所有任务;

[0193] b)9.0.0.2启动,触发第二轮投票并决策出由9.0.0.1、9.0.0.2执行所有任务;

[0194] c)9.0.0.3启动,触发第三轮投票并决策出由9.0.0.1、9.0.0.2、9.0.0.3执行所有任务;

[0195] d)9.0.0.1故障,触发第四轮投票并决策出由9.0.0.2、9.0.0.3执行所有任务。

[0196] 具体的,第一轮投票中,仅有ip为9.0.0.1的节点存活,则ip为9.0.0.1的进行第一轮投票,得到的第一轮投票结果为9.0.0.1为目标节点。

[0197] 在第二轮投票中,ip为9.0.0.1和ip为9.0.0.2的节点存活,则此时ip为9.0.0.1的节点在第二轮的投票结果为9.0.0.1和9.0.0.2,而ip为9.0.0.2的节点在第二轮投票结果为9.0.0.1和9.0.0.2。对于ip为9.0.0.1的节点来说,其能够获取到ip为9.0.0.2在第二轮的投票结果,则对于ip为9.0.0.1的节点来说,满足投票停止条件时会停止投票。对于ip为9.0.0.2的节点来说,其能够获取到ip为9.0.01在第二轮的投票结果,则对于ip为9.0.0.2

的节点来说也会停止投票,则此时各节点停止投票,则说明第二轮投票后ip为9.0.0.1的节点和ip为9.0.0.2的节点作为目标节点。

[0198] 在第三轮投票中,ip为9.0.0.1、ip为9.0.0.2和ip为9.0.0.3的节点存活。则此时ip为9.0.0.1的节点在第三轮的投票结果为9.0.0.1、9.0.0.2和9.0.0.3,而ip为9.0.0.2的节点在第三轮投票结果为9.0.0.1、9.0.0.2和9.0.0.3,ip为9.0.0.3的节点在第三轮的投票结果为9.0.0.1、9.0.0.2和9.0.0.3。对于ip为9.0.0.1、ip为9.0.0.2和ip为9.0.0.3中的任一节点来说,其能够获取到其他两个节点的投票结果,则ip为9.0.0.1、ip为9.0.0.2和ip为9.0.0.3中的任一节点会停止投票,则说明第三轮投票后ip为9.0.0.1的节点、ip为9.0.0.2和ip为9.0.0.3的节点作为目标节点。

[0199] 而到了第四轮投票,9.0.0.1故障,则ip为9.0.0.2和ip为9.0.0.3的节点存活。ip为9.0.0.2的节点在第四轮投票结果为9.0.0.2和9.0.0.3,ip为9.0.0.3的节点在第四轮的投票结果为9.0.0.2和9.0.0.3,则说明第四轮投票后ip为9.0.0.2和9.0.0.3的节点作为目标节点。

[0200] 请参阅表3,表3为本申请实施例提供的一种参与处理任务的目标节点列表。

[0201] 表3

[0202]

ballot	voter	vote	vote Time
1	9.0.0.1	9.0.0.1	2021-09-10 17:00:03
2	9.0.0.2	9.0.0.1;9.0.0.2	2021-09-10 17:01:03
2	9.0.0.1	9.0.0.1;9.0.0.2	2021-09-10 17:01:13
3	9.0.0.3	9.0.0.1;9.0.0.3	2021-09-10 17:02:23
3	9.0.0.1	9.0.0.1;9.0.0.2;9.0.0.3	2021-09-10 17:02:23
3	9.0.0.2	9.0.0.1;9.0.0.2;9.0.0.3	2021-09-10 17:02:33
4	9.0.0.3	9.0.0.2;9.0.0.3	2021-09-10 17:03:03
4	9.0.0.2	9.0.0.2;9.0.0.3	2021-09-10 17:03:03

[0203] 其中,如表3所示的演示过程中,区别在于第三轮投票中,9.0.0.3节点的投票结果为9.0.0.1和9.0.0.3,与9.0.0.1节点和9.0.0.2节点的投票结果均不同,则此时第三轮投票不满足投票结束条件,也就是说第三轮投票无法确定出目标节点,则进入第四轮投票。

[0204] 在一种可能的实现方式中,多个执行节点中的任一节点与投票管理数据库连接,投票管理数据库用于记录各执行节点每一轮投票处理得到的投票结果;

[0205] 查询至少一个第二节点分别对应的第N轮第二投票结果,包括:

[0206] 从投票管理数据库中查询至少一个第二节点分别对应的第N轮第二投票结果。

[0207] 在本实施例中,多个执行节点中的任一节点与投票管理数据库连接,则多个任务节点之间可以不需要进行通信也可以获取别的节点的投票结果,由此可以减少由于任务节点之间的通信数据丢失而导致的脑裂情况。

[0208] 在一种可能的实现方式中,投票管理数据库还用于记录各执行节点上报的心跳信息;

[0209] 获取各第二节点的心跳信息,包括:

[0210] 从投票管理数据库中获取各第二节点的心跳信息。

[0211] 在本实施例中,多个执行节点中的任一节点与投票管理数据库连接,则多个任务节点之间可以不需要进行通信也可以获取别的节点的心跳信息,由此可以减少由于任务节点之间的通信数据丢失而导致的脑裂情况。

[0212] 需要说明的是,本实施例中的投票管理数据库可以为一个,也可以为多个,在此不做限定。具体的,若投票管理数据库为多个,则多个投票管理数据库可以包括第一投票管理数据库和第二投票管理数据库,第一投票管理数据库用于记录各执行节点每一轮投票处理得到的投票结果,而第二投票管理数据库用于记录各执行节点上报的心跳信息。

[0213] 可以理解的是,通过第一投票管理数据库用于记录各执行节点每一轮投票处理得到的投票结果,而第二投票管理数据库用于记录各执行节点上报的心跳信息,则其中一个投票管理数据库异常时也能保留部分功能的使用。

[0214] 请参阅图6,图6为本申请实施例提供的一种节点的状态更新流程示意图。

[0215] 该流程的核心逻辑如下:

[0216] 1)每次心跳更新成功后,worker(节点)可继续工作work_timeout(10s);

[0217] 2)worker的心跳有server_deadline(15s)没更新后,表示该worker不存活;

[0218] 3)定时检查自己从心跳表看到的存活worker跟最后一次投票的存活worker是否一致,如不一致则发起新一轮投票;

[0219] 4)当实际存活worker列表和最后一次投票结果不同时,每个worker重新发起投票,到投票达成一致前,停止执行任务;

[0220] 5)投票达成一致的条件是:所有投票结果中的存活worker都参与了投票,并且投票结果相同;

[0221] 按照该流程更新执行节点的状态机,可能会导致某段时间内有任务没被执行,但可以保证任意时刻,各worker执行的任务没有交集。并且由于每个worker始终根据worker心跳表里存活的worker来作为投票结果,所以投票结果最终是收敛的。

[0222] 请参阅图7,图7为本申请实施例提供的一种分布式系统的任务处理方法的流程示意图。如图7所示的方法可以包括:

[0223] S710、确定多个执行节点中用于处理任务的目标节点。

[0224] 其中,目标节点可以是基于如上述任一方法实施例确定的,在此不做限定。

[0225] S720、若目标节点包括第一节点,则确定待处理任务中的第一数量个目标任务,并

对第一数量个目标任务进行处理。

[0226] 本实施例的技术方案,通过若满足投票触发条件,则进行第N轮投票处理,以得到第N轮第一投票结果,第一投票结果用于指示多个执行节点中处理任务的第一候选节点,基于第N轮第一投票结果确定多个执行节点中处理任务的目标节点,而每一轮投票处理,包括:获取各第二节点的心跳信息,并基于各第二节点的心跳信息确定第二节点中的存活节点,第二节点为多个执行节点中除第一节点外的节点,存活节点为最近一次心跳的时间与当前时间之间的时间间隔小于时间间隔阈值的节点;基于第二节点中的存活节点确定结果进行投票,得到第一投票结果,也就是说,在每一轮投票时,可以基于当前存活的节点来确定出用于处理任务的目标节点,从而持续更新处理任务的目标节点,由此可以解决分布式技术中容易出现脑裂的技术问题,从而达到减少脑裂出现的情况的技术效果。

[0227] 在一种可能的实现方式中,若目标节点为至少两个,则确定待处理任务中的第一数量个目标任务,包括以下的任一项:

[0228] 确定目标节点的第二数量,并基于待处理任务的任务总数量和第二数量确定第一节点处理的第一数量个任务,其中,第一数量与第二数量负相关,第一数量与任务总数量正相关;

[0229] 获取各目标节点的处理能力指示信息,并基于各处理能力指示信息所指示的任务处理能力确定各目标节点对应的比值系数,以及基于待处理任务的任务总数量和第一节点对应的比值系数确定第一节点处理的第一数量个任务,其中,各目标节点对应的比值系数与各目标节点对应的任务处理能力正相关,第一数量分别与第一节点对应的比值系数和任务总数量正相关。

[0230] 可选的,可以将任务总数量和第二数量的比值,确定为第一数量。具体的,当决策出参与处理任务的节点列表时,任务总分片数为节点列表长度,各节点执行的分片序号为按固定顺序(如节点id的字符序)排序的序号。假设某轮投票结束后,执行节点列表为:9.0.0.2、9.0.0.3,则任务总分片数为2,9.0.0.2执行序号为0的任务分片,9.0.0.3执行序号为1的任务分配,其中序号0的任务分片代表的任务集为:所有任务id%2=0的任务。

[0231] 处理能力指示信息可以是中央处理器(CPU)频率或核心数中的至少一项。

[0232] 本实施例的技术方案中,可以实现执行节点之间负载均衡,实现横向扩容的能力。

[0233] 请参阅图8,图8为本申请实施例提供的一种两个执行节点的CPU使用率的对比示意图。如图8所示,两个执行节点的cpu使用率是基本相同的。

[0234] 请参阅图9,图9为本申请实施例提供的一种分布式系统的节点确定装置的结构示意图。如图9所示的装置可以包括决策模块910。其中:

[0235] 决策模块910,用于若满足投票触发条件,则进行第N轮投票处理,以得到第N轮第一投票结果,第一投票结果用于指示多个执行节点中处理任务的第一候选节点,N为不小于1的整数;

[0236] 决策模块910还用于基于第N轮第一投票结果确定多个执行节点中处理任务的目标节点;

[0237] 其中,决策模块910在进行每一轮投票处理时,用于:

[0238] 获取各第二节点的心跳信息,并基于各第二节点的心跳信息确定第二节点中的存活节点,第二节点为多个执行节点中除第一节点外的节点,存活节点为最近一次心跳的时

间与当前时间之间的时间间隔小于时间间隔阈值的节点;

[0239] 基于第二节点中的存活节点确定结果进行投票,得到第一投票结果。

[0240] 可选的,投票触发条件包括以下的至少一项:

[0241] 到达投票触发时间;

[0242] 第一节点的最近一次心跳的时间与当前时间之间的时间间隔小于时间间隔阈值;

[0243] 多个执行节点中的存活节点与第N-1轮投票所确定的目标节点不一致,N为不小于2的整数;

[0244] 检测到第二节点中的存活节点发生变化。

[0245] 可选的,第一节点处于工作状态或暂停状态,处于工作状态的第一节点允许处理任务,处于暂停状态的第一节点禁止处理任务;决策模块910还用于:

[0246] 若确定出的目标节点包括第一节点,则将第一节点的状态设置为工作状态;

[0247] 若第一节点的最近一次心跳的时间与当前时间之间的时间间隔大于或等于时间间隔阈值,则将第一节点的状态设置为暂停状态;

[0248] 若满足投票触发条件且在确定出目标节点之前,将第一节点的状态设置为暂停状态。

[0249] 可选的,决策模块910在基于第N轮第一投票结果确定多个执行节点中处理任务的目标节点时,可以用于:

[0250] 查询至少一个第二节点分别对应的第N轮第二投票结果,第二投票结果用于指示多个执行节点中处理任务的第二候选节点;

[0251] 若第N轮第二投票结果查询失败,则将第N轮第一投票结果所指示的第一候选节点确定为目标节点;

[0252] 若第N轮第二投票结果查询成功,则基于第N轮第一投票结果和第N轮第二投票结果确定多个执行节点中处理任务的目标节点。

[0253] 可选的,基于第N轮第一投票结果和第N轮第二投票结果确定多个执行节点中处理任务的目标节点时,可以用于:

[0254] 若基于第N轮第一投票结果和第N轮第二投票结果确定满足投票结束条件,则确定用于处理任务的目标节点集合,并将目标节点集合中的任一节点确定为目标节点,目标节点集合为第N轮第一投票结果所指示的第一候选节点和第N轮第二投票结果所指示的第二候选节点之间的交集或并集;

[0255] 其中,投票结束条件包括以下的至少一项:

[0256] 第N轮第一投票结果与各第二节点对应的第N轮第二投票结果一致;

[0257] 查询到的第N轮第二投票结果所对应的至少一个第二节点与第二节点中的存活节点一致;

[0258] 方法还包括:

[0259] 若基于第N轮第一投票结果和第N轮第二投票结果确定不满足投票结束条件,则进行第N+1轮投票处理。

[0260] 可选的,多个执行节点中的任一节点与投票管理数据库连接,投票管理数据库用于记录各执行节点每一轮投票处理得到的投票结果;

[0261] 决策模块910在查询至少一个第二节点分别对应的第N轮第二投票结果时,可以用

于：

[0262] 从投票管理数据库中查询至少一个第二节点分别对应的第N轮第二投票结果。

[0263] 可选的，投票管理数据库还用于记录各执行节点上报的心跳信息；决策模块910在获取各第二节点的心跳信息时，可以用于：

[0264] 从投票管理数据库中获取各第二节点的心跳信息。

[0265] 可选的，决策模块910在基于第二节点中的存活节点确定结果进行投票，得到第一投票结果时，可以用于以下的至少一项：

[0266] 若存活节点确定结果为第二节点中的存活节点确定成功，则第一投票结果所指示的第一候选节点包括第一节点和第二节点中的存活节点；

[0267] 若存活节点确定结果为第二节点中的存活节点确定失败，则第一投票结果所指示的第一候选节点包括第一节点且不包括第二节点中的存活节点。

[0268] 请参阅图10，图10为本申请实施例提供的一种分布式系统的任务处理装置的结构示意图。如图10所示的装置可以包括决策模块910和执行模块920，其中：

[0269] 决策模块910，用于确定多个执行节点中用于处理任务的目标节点，其中，目标节点是基于如以上任一实施例的方法确定的；

[0270] 执行模块920，用于若目标节点包括第一节点，则确定待处理任务中的第一数量个目标任务，并对第一数量个目标任务进行处理。

[0271] 可选的，若目标节点为至少两个，则执行模块920在确定待处理任务中的第一数量个目标任务，可以用于以下的任一项：

[0272] 确定目标节点的第二数量，并基于待处理任务的任务总数量和第二数量确定第一节点处理的第一数量个任务，其中，第一数量与第二数量负相关，第一数量与任务总数量正相关；

[0273] 获取各目标节点的处理能力指示信息，并基于各处理能力指示信息所指示的任务处理能力确定各目标节点对应的比值系数，以及基于待处理任务的任务总数量和第一节点对应的比值系数确定第一节点处理的第一数量个任务，其中，各目标节点对应的比值系数与各目标节点对应的任务处理能力正相关，第一数量分别与第一节点对应的比值系数和任务总数量正相关。

[0274] 本申请实施例的装置可执行本申请实施例所提供的方法，其实现原理相类似，本申请各实施例的装置中的各模块所执行的动作是与本申请各实施例的方法中的步骤相对应的，对于装置的各模块的详细功能描述具体可以参见前文中所示的对应方法中的描述，此处不再赘述。

[0275] 本申请实施例中提供了一种执行节点，包括存储器、处理器及存储在存储器上的计算机程序，该处理器执行上述计算机程序以实现任一实施例的方法的步骤。

[0276] 在一个可选实施例中提供了一种执行节点，如图11所示，图11所示的执行节点1100包括：处理器1101和存储器1103。其中，处理器1101和存储器1103相连，如通过总线1102相连。可选地，执行节点1100还可以包括收发器1104，收发器1104可以用于该执行节点与其他执行节点之间的数据交互，如数据的发送和/或数据的接收等。需要说明的是，实际应用中收发器1104不限于一个，该执行节点1100的结构并不构成对本申请实施例的限定。

[0277] 处理器1101可以是CPU(Central Processing Unit，中央处理器)，通用处理器，

DSP(Digital Signal Processor,数据信号处理器),ASIC(Application Specific Integrated Circuit,专用集成电路),FPGA(Field Programmable Gate Array,现场可编程门阵列)或者其他可编程逻辑器件、晶体管逻辑器件、硬件部件或者其任意组合。其可以实现或执行结合本申请公开内容所描述的各种示例性的逻辑方框,模块和电路。处理器1101也可以是实现计算功能的组合,例如包含一个或多个微处理器组合,DSP和微处理器的组合等。

[0278] 总线1102可包括一通路,在上述组件之间传送信息。总线1102可以是PCI(Peripheral Component Interconnect,外设部件互连标准)总线或EISA(Extended Industry Standard Architecture,扩展工业标准结构)总线等。总线1102可以分为地址总线、数据总线、控制总线等。为便于表示,图11中仅用一条粗线表示,但并不表示仅有一根总线或一种类型的总线。

[0279] 存储器1103可以是ROM(Read Only Memory,只读存储器)或可存储静态信息和指令的其他类型的静态存储设备,RAM(Random Access Memory,随机存取存储器)或者可存储信息和指令的其他类型的动态存储设备,也可以是EEPROM(Electrically Erasable Programmable Read Only Memory,电可擦可编程只读存储器)、CD-ROM(Compact Disc Read Only Memory,只读光盘)或其他光盘存储、光碟存储(包括压缩光碟、激光碟、光碟、数字通用光碟、蓝光光碟等)、磁盘存储介质、其他磁存储设备、或者能够用于携带或存储计算机程序并能够由计算机读取的任何其他介质,在此不做限定。

[0280] 存储器1103用于存储执行本申请实施例的计算机程序,并由处理器1101来控制执行。处理器1101用于执行存储器1103中存储的计算机程序,以实现前述方法实施例所示的步骤。

[0281] 可选的,执行节点可以包括但不限于是个人计算机、笔记本电脑、智能手机、平板电脑、物联网设备、便携式可穿戴设备或服务器中的至少一种,物联网设备可为智能音箱、智能电视、智能空调或智能车载设备中的至少一种。便携式可穿戴设备可为智能手表、智能手环或头戴设备等中的 一种。服务器可以用独立的服务器或者是多个服务器组成的服务器集群来实现。

[0282] 本申请实施例提供了一种分布式系统,包括多个执行节点,其中,多个执行节点的任一节点作为第一节点执行以上任一实施例的方法。

[0283] 本申请实施例提供了一种计算机可读存储介质,该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可实现前述方法实施例的步骤及相应内容。

[0284] 本申请实施例还提供了一种计算机程序产品,包括计算机程序,计算机程序被处理器执行时可实现前述方法实施例的步骤及相应内容。

[0285] 应该理解的是,虽然本申请实施例的流程图中通过箭头指示各个操作步骤,但是这些步骤的实施顺序并不受限于箭头所指示的顺序。除非本文中有明确的说明,否则在本申请实施例的一些实施场景中,各流程图中的实施步骤可以按照需求以其他的顺序执行。此外,各流程图中的部分或全部步骤基于实际的实施场景,可以包括多个子步骤或者多个阶段。这些子步骤或者阶段中的部分或全部可以在同一时刻被执行,这些子步骤或者阶段中的每个子步骤或者阶段也可以分别在不同的时刻被执行。在执行时刻不同的场景下,这些子步骤或者阶段的执行顺序可以根据需求灵活配置,本申请实施例对此不限制。

[0286] 以上仅是本申请部分实施场景的可选实施方式,应当指出,对于本技术领域的普通技术人员来说,在不脱离本申请的方案技术构思的前提下,采用基于本申请技术思想的其他类似实施手段,同样属于本申请实施例的保护范畴。

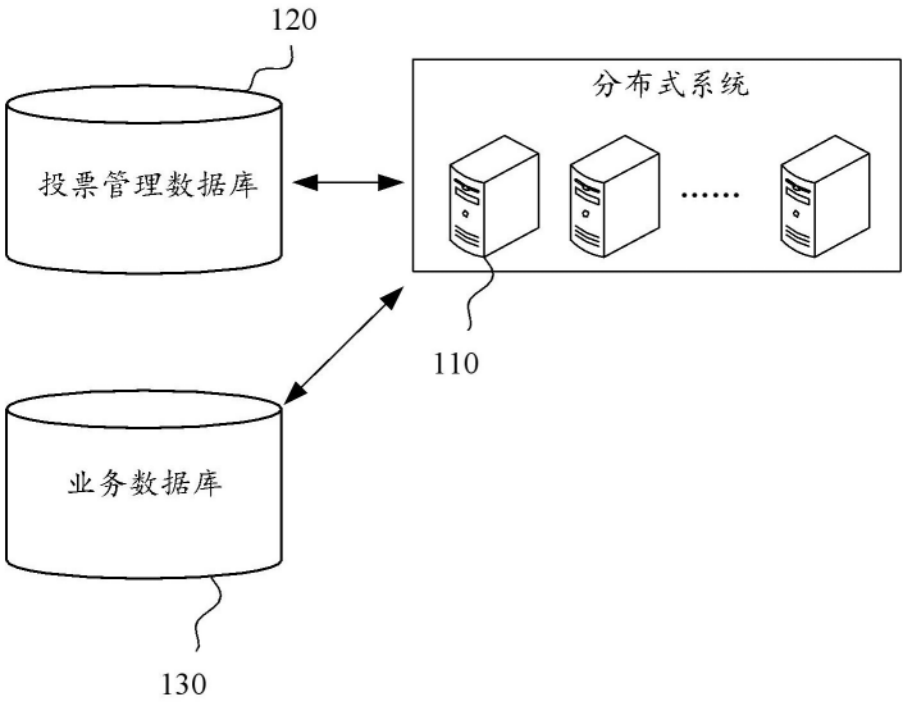

120
分布式系统
投票管理数据库
.....
110
业务数据库
130

图1

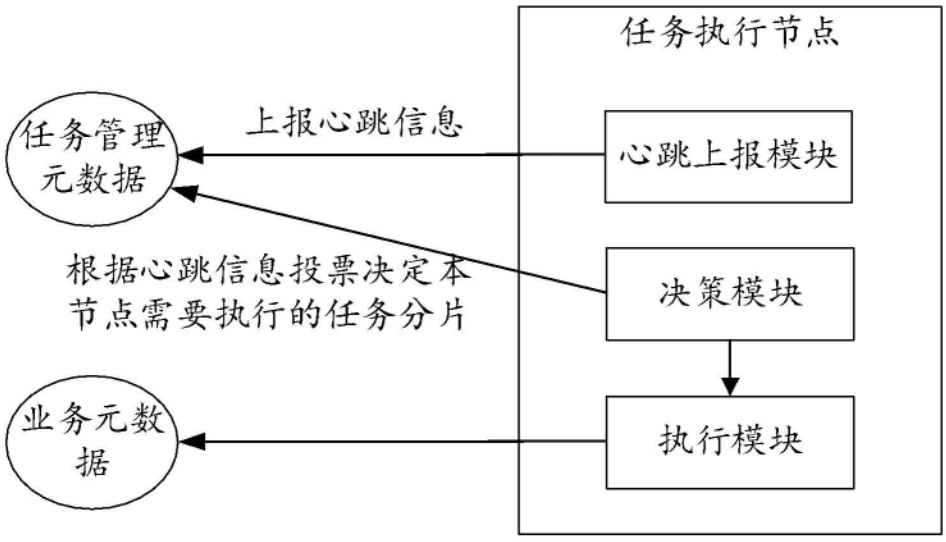

任务执行节点
任务管理元数据
上报心跳信息
心跳上报模块
根据心跳信息投票决定本节点需要执行的任务分片
决策模块
执行模块
业务元数据

图2

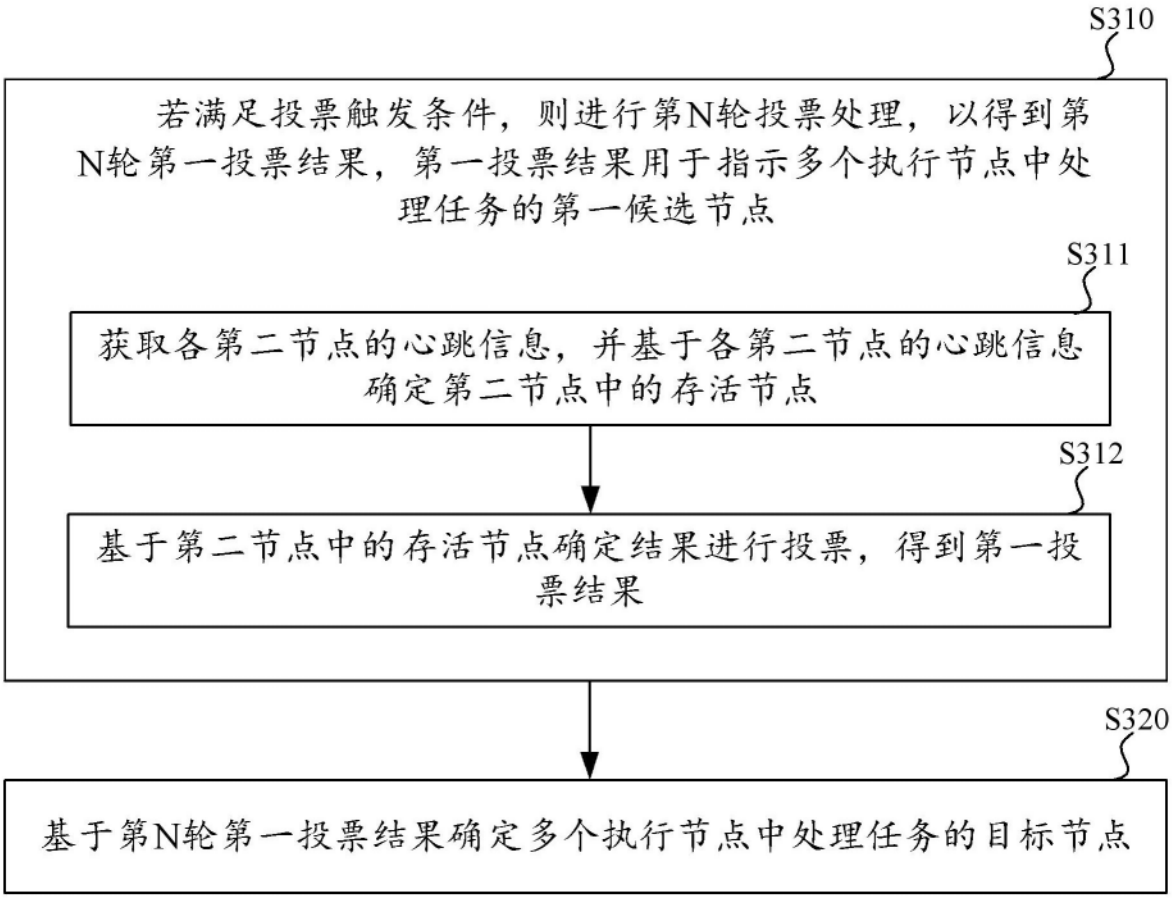

S310
若满足投票触发条件，则进行第N轮投票处理，以得到第N轮第一投票结果，第一投票结果用于指示多个执行节点中处理任务的第一候选节点
S311
获取各第二节点的心跳信息，并基于各第二节点的心跳信息确定第二节点中的存活节点
S312
基于第二节点中的存活节点确定结果进行投票，得到第一投票结果
S320
基于第N轮第一投票结果确定多个执行节点中处理任务的目标节点

图3

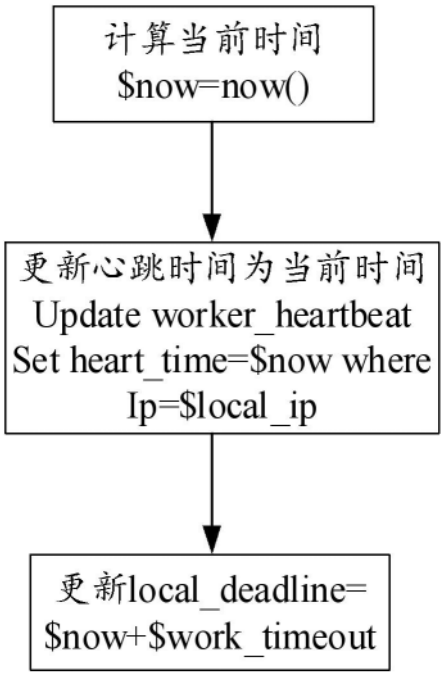

计算当前时间
$now=now()
更新心跳时间为当前时间
Update worker_heartbeat
Set heart_time=$now where
Ip=$local_ip
更新local_deadline=
$now+$work_timeout

图4

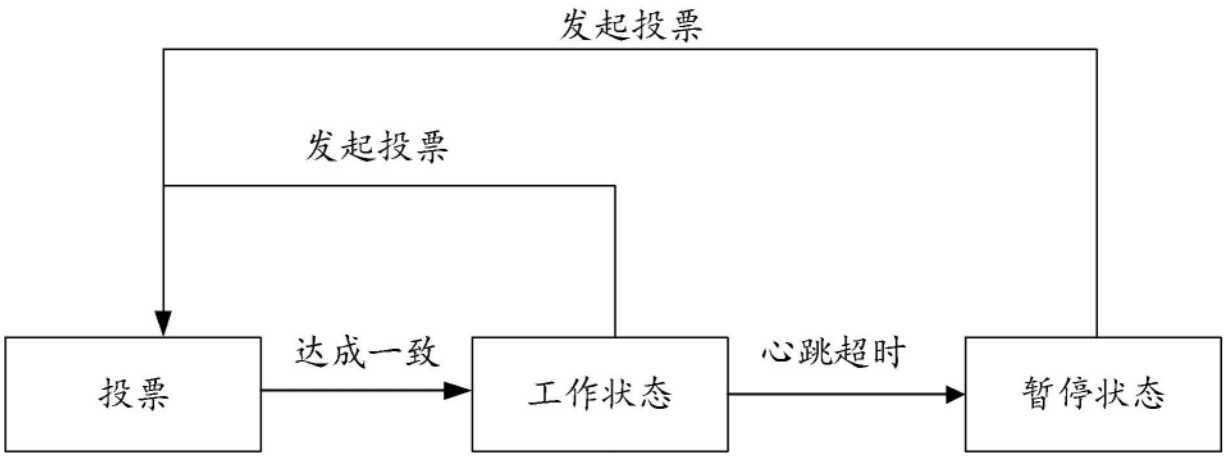

发起投票
发起投票
投票
达成一致
工作状态
心跳超时
暂停状态

图5

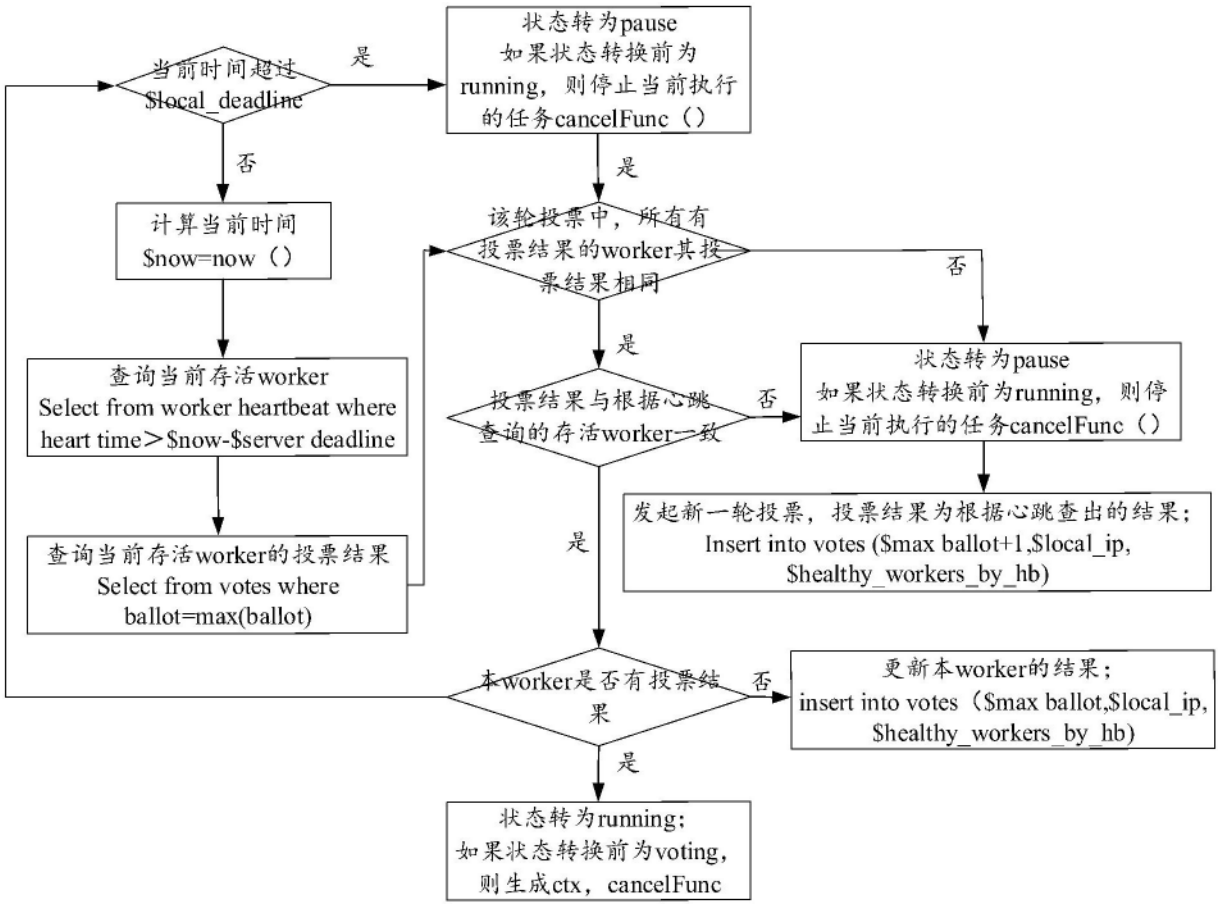

当前时间超过$local_deadline
是
否
状态转为pause
如果状态转换前为running，则停止当前执行的任务cancelFunc（）
是
计算当前时间
$now=now（）
该轮投票中，所有有投票结果的worker其投票结果相同
否
是
查询当前存活worker
Select from worker heartbeat where heart time＞$now-$server deadline
投票结果与根据心跳查询的存活worker一致
否
状态转为pause
如果状态转换前为running，则停止当前执行的任务cancelFunc（）
查询当前存活worker的投票结果
Select from votes where ballot=max(ballot)
是
发起新一轮投票，投票结果为根据心跳查出的结果；
Insert into votes ($max ballot+1,$local_ip, $healthy_workers_by_hb)
本worker是否有投票结果
否
更新本worker的结果；
insert into votes（$max ballot,$local_ip, $healthy_workers_by_hb)
是
状态转为running;
如果状态转换前为voting，则生成ctx，cancelFunc

图6

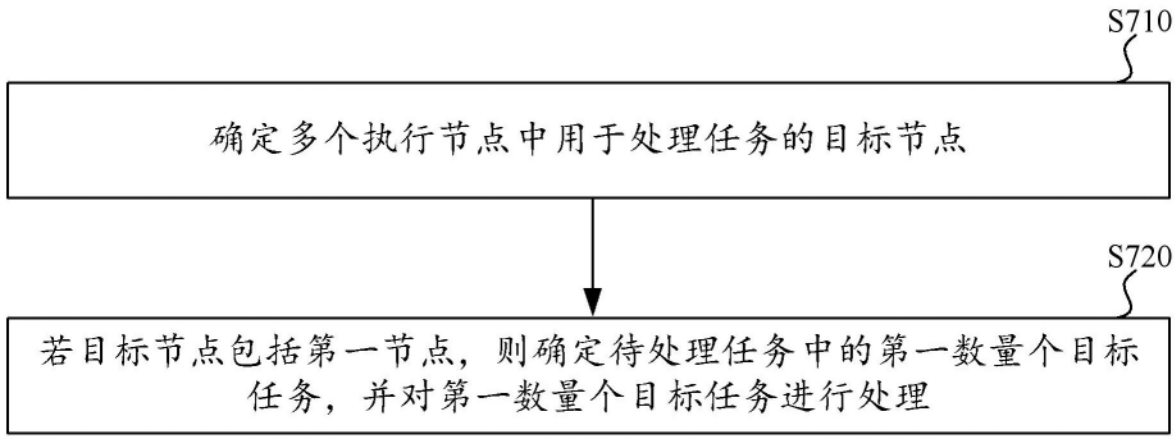

S710
确定多个执行节点中用于处理任务的目标节点
S720
若目标节点包括第一节点，则确定待处理任务中的第一数量个目标任务，并对第一数量个目标任务进行处理

图7

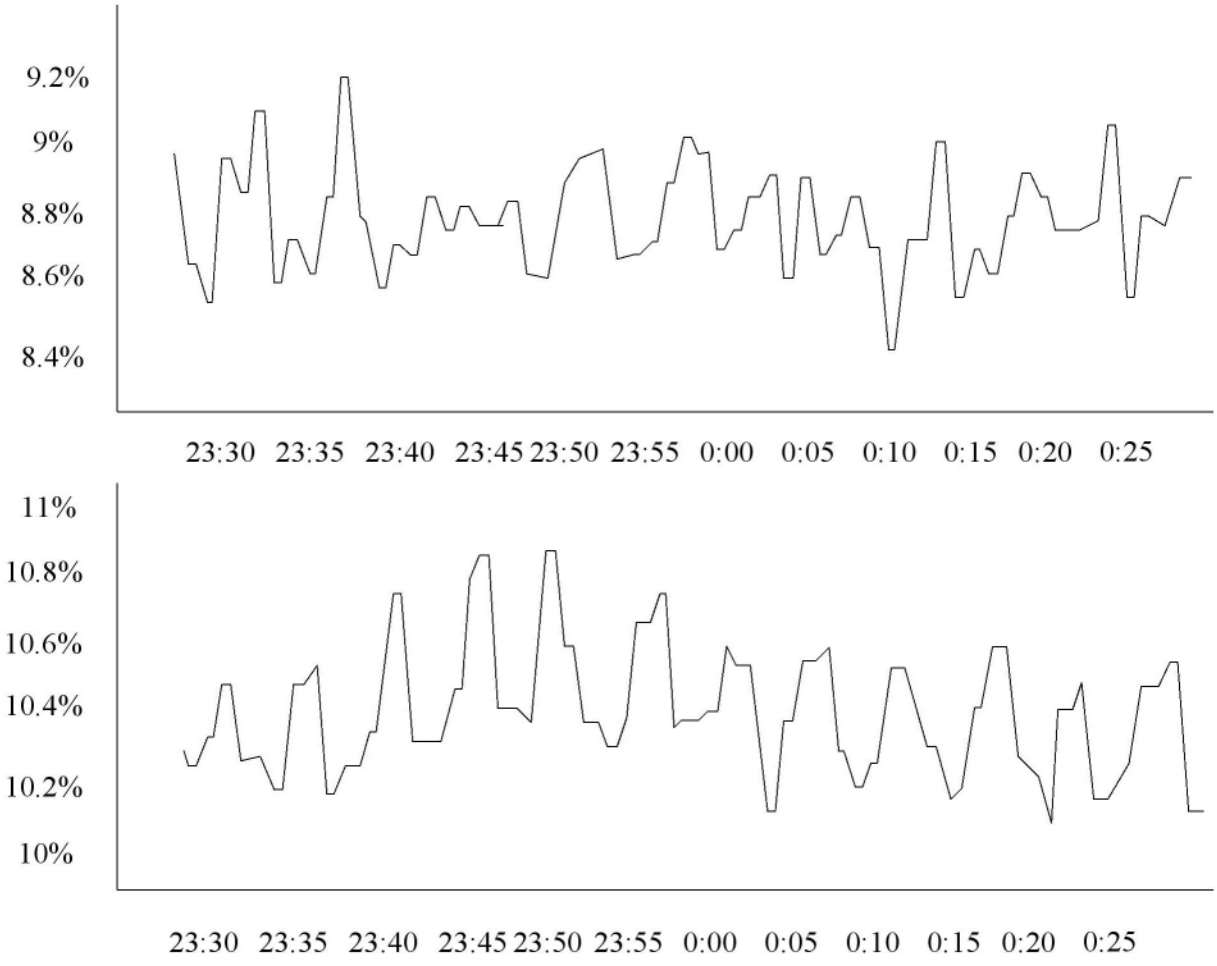

9.2%
9%
8.8%
8.6%
8.4%
23:30 23:35 23:40 23:45 23:50 23:55 0:00 0:05 0:10 0:15 0:20 0:25
11%
10.8%
10.6%
10.4%
10.2%
10%
23:30 23:35 23:40 23:45 23:50 23:55 0:00 0:05 0:10 0:15 0:20 0:25

图8

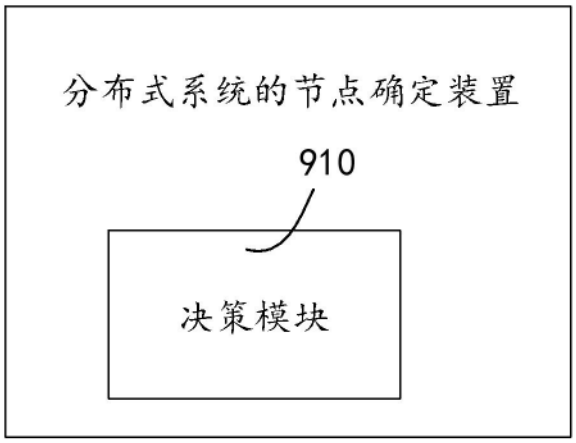

分布式系统的节点确定装置
910
决策模块

图9

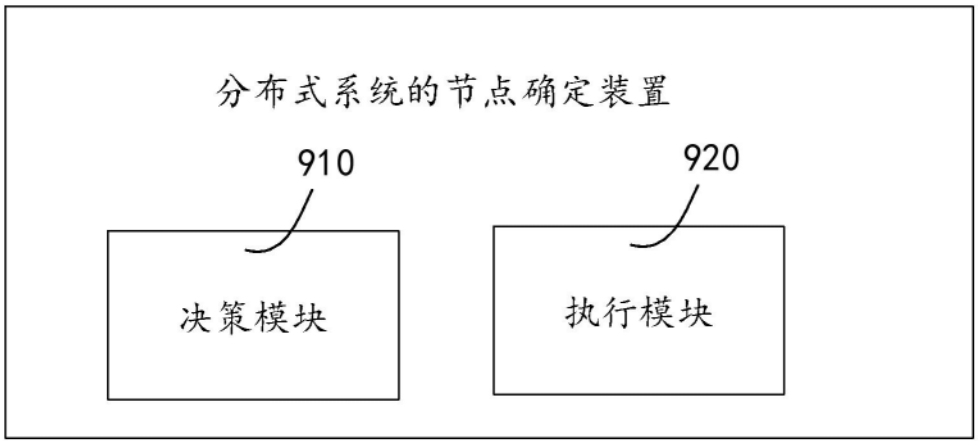

分布式系统的节点确定装置
910
决策模块
920
执行模块

图10

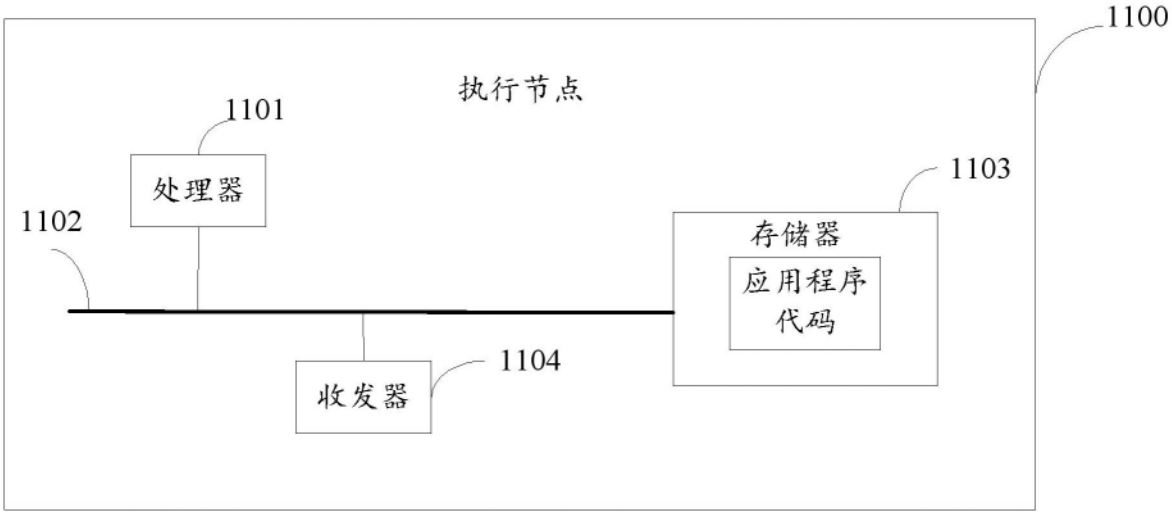

1100
执行节点
1101
处理器
1102
1103
存储器
应用程序
代码
1104
收发器

图11