



(12) 发明专利

(10) 授权公告号 CN 110113406 B

(45) 授权公告日 2022. 04. 08

(21) 申请号 201910356130.1

(22) 申请日 2019.04.29

(65) 同一申请的已公布的文献号
申请公布号 CN 110113406 A

(43) 申请公布日 2019.08.09

(73) 专利权人 成都网阔信息技术股份有限公司
地址 610041 四川省成都市成都高新区天府二街萃华路89号33层

(72) 发明人 刘宏基

(74) 专利代理机构 成都弘毅天承知识产权代理有限公司 51230

代理人 邹敏菲

(51) Int. Cl.

H04L 67/10 (2022.01)

H04L 12/42 (2006.01)

(56) 对比文件

CN 109584106 A, 2019.04.05

CN 106357449 A, 2017.01.25

CN 109327509 A, 2019.02.12

US 2016301624 A1, 2016.10.13

CN 106131213 A, 2016.11.16

审查员 郑骏

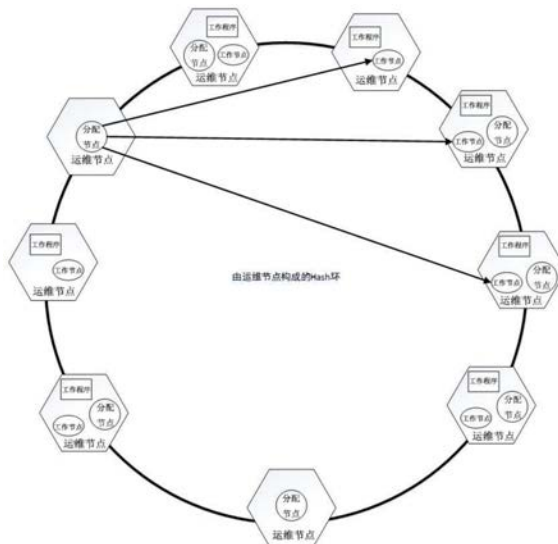
权利要求书2页 说明书9页 附图5页

(54) 发明名称

基于分布式的计算服务集群系统

(57) 摘要

本发明公开了基于分布式的计算服务集群框架,将框架结构设计为环状巧妙解决节点的主备问题,使得各个节点处于等效位置,框架的高效性、健壮性得到提升;完全自主开发,不依赖任何第三方案程序,代码量少,容易理解掌握;增加任务请求跟踪模式解决了现有框架的实时性不高的问题;运维节点、分配节点、工作节点处理逻辑均与具体业务不相关,而是将具体业务处理放在由客户自主开发的工作程序中,大大提升同一套计算框架对数据类型的适应性。本计算框架可以有效整合闲散、低效计算资源,输出高效的计算能力,提高已有计算资源的利用率,为构建绿色环保节约型社会贡献力量。



1. 基于分布式的计算服务集群系统,包括至少两个部署了二进制程序的物理机,其特征在于:

在所述物理机上还部署有一个运维节点、一个分配节点、一个工作节点和至少一个工作程序;

所有物理机上的运维节点组成一个Hash环;

还包括由自定义模块层、框架模块层和网络层组成的逻辑结构;

所述网络层包括网络库C++接口、网络库Python接口、网络库C接口和Python通用库中的至少一个;

所述框架模块层包括运维节点模块、分配节点模块和工作节点模块,还包括与运维节点模块、分配节点模块和工作节点模块均连接的数据结构模块、线程函数模块、过程模块、协议模块和Util模块;

所述自定义模块层包括工作程序模块和客户端模块;

所述各个运维节点的功能相同,每个运维节点加入系统后在本地保存一张全表信息,包含系统上所有运维节点的状态信息,在节点被删除或出现异常时其他节点的全表信息会得到更新,每个节点通过例测线程检测前后相连节点的健康状态并定期做全表信息同步,信息表中多节点在环中的位置是按加入顺序生成的Hashid固定的。

2. 根据权利要求1所述的基于分布式的计算服务集群系统的任务处理方法,其特征在于:包括如权利要求1所述的计算服务集群框架,还包括一个装有与计算服务集群框架匹配的客户端程序的客户端,还包括以下步骤:

S1、客户端向集群维护者申请调用账号、密码;客户端程序通过账号、密码向运维节点发起登录请求;

S2、运维节点验证客户的程序发送的账号、密码,验证成功后运维节点获取分配节点信息,返回分配信息到客户端;

S3、客户端根据运维节点返回的分配信息链接分配节点;

S4、客户端发送任务请求到分配节点,分配节点根据实际的任务类型和下属的工作节点负载情况综合配置任务请求的分割方式和接受任务的工作节点,然后将任务转发到接收任务的工作节点;

S5、接收到任务的工作节点调取对应的工作程序,将任务转发到工作程序进行处理,工作程序完成任务后将任务结果返回到工作节点;

S6、工作节点将任务结果返回分配该任务的分配节点;

S7、接收到任务结果的分配节点将任务结果返回到发送任务请求的客户端。

3. 根据权利要求2所述的基于分布式的计算服务集群系统的任务处理方法,其特征在于:所述步骤S5中,接收到任务的工作节点为任务进行编号,并将编号和发送任务的分配节点匹配后进行记录,在步骤S6中,工作节点通过读取任务结果的任务编号,通过任务编号匹配发送任务的分配节点,将任务结果返回该分配节点。

4. 根据权利要求3所述的基于分布式的计算服务集群系统,其特征在于:所述运维节点在工作时包括主动过程任务和被动过程任务。

5. 根据权利要求4所述的基于分布式的计算服务集群系统,其特征在于:所述主动过程任务和被动过程任务均包括加入云请求、退出云请求、相邻节点健康检查、表信息收集、表

信息同步、新增节点、删除节点和获取信息表中的至少一个。

基于分布式的计算服务集群系统

技术领域

[0001] 本发明涉及分布式计算领域,特别涉及基于分布式的计算服务集群系统。

背景技术

[0002] 目前市面上有大量的专用计算服务框架,如批处理框架Hadoop,流处理框架Storm,以及混合处理型框架Spark。

[0003] Hadoop主要依赖MapReduce引擎实现批处理,采用分布式文件系统HDFS,靠对数据块的多副本备份来解决服务器宕机和磁盘损坏问题。

[0004] Storm会对实时进入系统的数据进行计算,很适合用来处理必须对变动或峰值做出响应并且需要关注一段时间内变化趋势的数据。Storm流处理框架可对其中名为Topology的DAG进行编排,可以指定每个传入的片段需要执行的不同步骤。

[0005] Spark是一种可以处理批处理任务的流处理框架,对一些特殊算法有一定针对性,另外Spark会对输入数据进行缓存,每次计算不必重新加载,对计算加速有非常大的促进作用。

[0006] 上述几种框架就是目前比较流行的计算服务框架,都是基于主从结构对框架展开的,在细节上不同分布式计算框架的结构则重点和应用场景都有所差异。

[0007] 批处理框架Hadoop严重依赖磁盘的持久存储,每个任务需要执行多次的读取和写入操作,速度较慢。在表层提供Map和Reduce两个操作,处理逻辑隐藏在代码中,整体逻辑不够清晰,处理数据时延性高,只适合批处理数据,难以处理实时数据,缺点是时延高,处理流程固定。

[0008] 流处理框架Storm很适合处理对延迟需求很高的纯粹流处理负载,但是无法满足批处理要求,如果需要批处理能力则还需要配合其他软件。缺点是依赖其他组件较多,内存控制不好,无法单独处理批数据。

[0009] 混合处理框架Spark基于对输入数据的缓存机制,需要较大的内存,同时对数据的增量更新效率差。

[0010] 总结市面现有分布式计算框架存在的技术缺点是均是基于主从结构,框架高效性有待提升。

发明内容

[0011] 本发明的目的在于:提供了基于分布式的计算服务集群系统,具有对数据类型适应性很强,数据处理的实时性很高,内存操作自主控制,内存消耗量小,同时不依赖任何第三方开源组件的特点,最重要的是本计算框架数据处理逻辑清晰,代码量小,容易掌握理解,呈环状的框架结构,有效提升了框架的整体稳定性和高效性,解决了现有分布式计算框架均是基于主从结构,框架高效性有待提升的问题

[0012] 本发明采用的技术方案如下:

[0013] 基于分布式的计算服务集群系统,包括至少两个部署了二进制程序的物理机,在

所述物理机上还部署有一个运维节点、一个分配节点、一个工作节点和至少一个工作程序；所有物理机上的运维节点组成一个Hash环。本架构前置条件包括：

[0014] 输入任务具备明确可划分的特性，才能充分显示分布式系统的优越性，但是不可划分任务在本框架中仍然适用；具备一套通用的异常数据处理机制；数据流每个级别都有对应的标记跟踪任务完成情况；本框架为计算密集型架构，各个节点或节点与客户端之间不应该具有大量数据的高速传输，数据通道和控制通道为全双工模式。

[0015] 框架拓扑结构为：

[0016] 物理结构：每台物理机上部署全套二进制程序，主要包括1个运维节点、1个分配节点、1个工作节点、n个工作程序。

[0017] 逻辑结构：每台物理机上运维节点为必须运行节点，主要负责加入系统云后对整个云系统的维护、数据同步、信息判断、仲裁等等。而分配节点、工作节点、工作程序在物理机上不一定必须运行，运行模式为在运维节点进行信息同步后决定启动哪些分配节点和工作节点，以及分配节点和工作节点之间如何形成归属关系，包括工作节点应该启动哪些工作程序。

[0018] 在线升级：采用failover/failback，升级时部分节点先升级，然后逐步全部升级，实现不停机不停服无感升级。

[0019] 逻辑节点定义：

[0020] 运维节点：运维节点主要承担任务均衡的工作，包括分配节点、工作节点、其他节点加入云后的归属关系均衡、客户初次加入云后的链接信息分配等。运维节点间为环状组织结构，数据信息在环上为顺序流动，同正向或同逆向。各个运维节点的重要性是平等的，没有主备之分，这是与其他同类集群框架的重要差异性之一。

[0021] 分配节点：分配节点主要承担直接与客户端通信，接收来自客户端的任务，并将一个完整的业务均衡到不同的工作节点上，追踪任务直到完成，达到提高任务完成速度的目的，也就是要具备任务分割和转发功能。分配节点的实现逻辑不能涉及具体的业务，同时应该具备全息性，单分配节点也能完成多分配节点能完成的任务。与客户端直接通信还应该具备接入身份验证、任务数限制功能。在控制指令方面原则上接收来自运维节点的指令，推送指令到客户端；而信息同步是推送信息到运维节点，接收工作节点的信息。

[0022] 工作节点：一台物理设备上运行一个工作节点，工作节点接收其所属分配节点的数据，映射任务到下属的工作程序，同时维护下属工作程序的负载情况并反馈给分配节点。工作节点需要能兼容下属工作程序的并发性质，包括无限制并发、有限限制并发和串行。另外工作节点需要维护一张描述工作程序属性的表，用于汇总工作节点下属工作程序的负载情况，并同时同步到所属的分配节点，作为分配节点负载均衡的判断依据。

[0023] 工作程序：工作程序对应一种或一类具体的业务，每个工作程序在分布式框架中对应一个具体的业务编号，作为进程独立存在。例如负责位置解析的工作程序只完成位置解析任务，负责视频分析的工作程序只完成视频分析任务。工作程序的通信协议必须遵从现有的分布式框架通信协议。

[0024] 客户端程序：客户端是用户直接调用的程序，一般以动态库等形式存在，为调用者提供具体的调用接口。客户端通常需要包括并发调用、接入合法性验证、简单负载均衡功能。

- [0025] 进一步的,包括上述的计算服务集群框架,还包括由自定义模块层、框架模块层和网络层组成的逻辑结构;
- [0026] 所述网络层包括网络库C++接口、网络库Python接口、网络库C接口和Python通用库中的至少一个;
- [0027] 所述框架模块层包括运维节点模块、分配节点模块和工作节点模块,还包括与运维节点模块、分配节点模块和工作节点模块均连接的数据结构模块、线程函数模块、过程模块、协议模块和Util模块;
- [0028] 数据结构模块:负责自定义数据结构和数据处理;
- [0029] 线程函数模块:负责实现各种线程函数,包括例测线程、任务线程、网络函数回调线程等;
- [0030] 过程模块:负责实现各种过程,包括源语映射等;
- [0031] 协议模块:负责控制协议和任务协议等各种协议数据的打包和解包工作;
- [0032] Util模块:包括实现其他功能的函数群。
- [0033] 自定义模块层:用户根据实际业务需求案子框架协议定义的功能。
- [0034] 所述自定义模块层包括工作程序模块和客户端模块。
- [0035] 网络层:主要负责提供整个框架所需的网络组建接口,包括多种语言接口;
- [0036] 框架模块层:构成框架必备模块,分为运维节点模块、分配节点模块和工作节点模块,上述三个模块还包括多个功能模块,功能模块包括数据结构模块、线程函数模块、过程模块、协议模块和Util模块,
- [0037] 基于分布式的计算服务集群系统的任务处理方法,包括上述的计算服务集群框架,还包括一个装有与计算服务集群框架匹配的客户端程序的客户端,还包括以下步骤:
- [0038] S1、客户端向集群维护者申请调用账号、密码;客户端程序通过账号、密码向运维节点发起登录请求;
- [0039] S2、运维节点验证客户的程序发送的账号、密码,验证成功后运维节点获取分配节点信息,返回分配信息到客户端;
- [0040] S3、客户端根据运维节点返回的分配信息链接分配节点;
- [0041] S4、客户端发送任务请求到分配节点,分配节点根据实际的任务类型和下属的工作节点负载情况综合配置任务请求的分割方式和接受任务的工作节点,然后将任务转发到接手任务的工作节点;
- [0042] S5、接收到任务的工作节点调取对应的工作程序,将任务转发到工作程序进行处理,工作程序完成任务后将任务结果返回到工作节点;
- [0043] S6、工作节点将任务结果返回分配该任务的分配节点;
- [0044] S7、接收到任务结果的分配节点将任务结果返回到发送任务请求的客户端。
- [0045] 分布式集群通常由n台物理设备组合而成,每台物理设备中部署全套共4种二进制程序,参照前面介绍的框架拓扑结构来部署。对外提供1个或多个集群入口地址,启动了分配节点的物理节点即为一个入口地址,整套系统中可以有多个分配节点被启动。
- [0046] 客户端调用者需要向集群维护者申请调用账号密码,通过账号密码首先向集群入口地址发起访问,由入口运维节点返回实际访问地址,这一步对调用者不可见。
- [0047] 认证通过后即可发起任务请求,客户端的任务请求到达分配节点后,由分配节点

根据实际的任务类型和下属的工作节点负载情况综合裁定任务请求的分割方式和接受任务的工作节点。

[0048] 工作节点接收到任务请求后根据请求类型和下属工作程序的负载情况决定接受任务的工作程序。工作程序在完成任务后根据请求类型按原来返回结果或者不返回。

[0049] 在整个数据处理流程中各个节点都会根据任务请求类型做出对应的响应,包括是否要等待任务完成、是否是按并发处理或串行处理等等,都是有一套完整的通信协议来控制的,本发明技术关键点不在通信协议,所以对通信协议不做详细阐述。

[0050] 为了维护整个集群框架的稳定正常运转,各个逻辑节点间的相互控制关系也是极其重要的。

[0051] 首先运维节点是重中之重,每个物理设备必须运行一个运维节点,由多个运维节点构成信息环,运维节点间的数据在环上顺序流动。运维节点主动向分配节点、工作节点推送控制指令,而分配节点、工作节点有信息同步需求则要主动推送到运维节点。

[0052] 进一步的,所述步骤S5中,接收到任务的工作节点为任务进行编号,并将编号和发送任务的分配节点匹配后进行记录,在步骤S6中,工作节点通过读取任务结果的任务编号,通过任务编号匹配发送任务的分配节点,将任务结果返回该分配节点。

[0053] 进一步的,所述运维节点在工作时包括主动过程任务和被动过程任务。

[0054] 主动过程任务为当前阶段主动向向量的前后阶段发送请求获取相应的信息;

[0055] 被动过程任务为当前节点接收到相邻节点的请求信息后作出相应的信息反馈。

[0056] 进一步的,所述主动过程任务和被动过程任务均包括加入云请求、退出云请求、相邻节点健康检查、表信息收集、表信息同步、新增节点、删除节点和获取信息表中的至少一个。

[0057] 进一步的,所述各个运维节点的功能相同,每个运维节点加入系统后在本地保存一张全表信息,包含系统上所有运维节点的状态等信息,在节点被删除或出现异常时其他节点的全表信息会得到更新,每个节点通过例测线程检测前后相连节点的健康状态并定期做全表信息同步,信息表中多节点在环中的位置是按加入顺序生成的Hashid固定的。

[0058] 基于前面对运维节点的功能介绍,可见框架对异常情况的处理就主要集中在运维节点中,而异常情况的处理合理性直接决定了整个框架的健壮性。

[0059] 关键点就在于运维节点的等效性,即是各个运维节点的功能相同,不存在常规分布式框架中的仲裁等机制,多运维节点环上的任何节点丢失都不影响系统的正常运行,具体保障机制如下所述:

[0060] 云管理过程:每个运维节点加入云后在本地保存一张全表信息,包含云上所有运维节点的状态等信息,在节点被删除或出现异常时其他节点的全表信息会得到更新,每个节点通过例测线程检测前后相连节点的健康状态并定期做全表信息同步,信息表中多节点在环中的位置是按加入顺序生成的Hashid固定的。详情可参见例图5。

[0061] 云均衡过程:每个运维节点负责控制本机工作节点/分配节点的启动和关闭,监测本机分配节点/工作节点的工作状态,通过一系列主动例测线程实现这一功能。

[0062] 通过上述保障机制在运维节点环中能够实现动态调整节点,达到整个环状结构中只要有一个节点正常运行,则整个系统就能正常对外提供服务,从而有效提升了系统的健壮性。

[0063] 综上所述,由于采用了上述技术方案,本发明的有益效果是:

[0064] 1. 本发明基于分布式的计算服务集群系统,将框架结构设计为环状巧妙解决节点的主备问题,使得各个节点处于等效位置,框架的高效性、健壮性得到提升;完全自主开发,不依赖任何第三程序,代码量少,容易理解掌握;增加任务请求跟踪模式解决了现有框架的实时性不高的问题;运维节点、分配节点、工作节点处理逻辑均与具体业务不相关,而是将具体业务处理放在由客户自主开发的工作程序中,大大提升同一套计算框架对数据类型的适应性;

[0065] 2. 本发明基于分布式的计算服务集群系统,同时由于本框架对计算资源的要求几乎为0,几乎可以适应所有现有工作PC机,如果是工作服务器那更是如虎添翼,所以本计算框架可以有效整合闲散、低效计算资源,输出高效的计算能力,提高已有计算资源的利用率,为构建绿色环保节约型社会贡献力量。

附图说明

[0066] 本发明将通过例子并参照附图的方式说明,其中:

[0067] 图1是本发明的框架整体拓扑结构;

[0068] 图2是本发明的框架逻辑视图;

[0069] 图3是本发明的框架任务流程视图;

[0070] 图4是本发明的框架各个逻辑节点间的相互控制关系视图;

[0071] 图5是本发明的框架运维节点云管理过程视图。

具体实施方式

[0072] 本说明书中公开的所有特征,或公开的所有方法或过程中的步骤,除了互相排斥的特征和/或步骤以外,均可以以任何方式组合。

[0073] 下面结合图1至图5对本发明作详细说明。

[0074] 实施例1

[0075] 如图1所示,基于分布式的计算服务集群系统,包括至少两个部署了二进制程序的物理机,在所述物理机上还部署有一个运维节点、一个分配节点、一个工作节点和至少一个工作程序;所有物理机上的运维节点组成一个Hash环。本架构前置条件包括:

[0076] 输入任务具备明确可划分的特性,才能充分显示分布式系统的优越性,但是不可划分任务在本框架中仍然适用;具备一套通用的异常数据处理机制;数据流每个级别都有对应的标记跟踪任务完成情况;本框架为计算密集型架构,各个节点或节点与客户端之间不应该具有大量数据的高速传输,数据通道和控制通道为全双工模式。

[0077] 框架拓扑结构为:

[0078] 物理结构:每台物理机上部署全套二进制程序,主要包括1个运维节点、1个分配节点、1个工作节点、n个工作程序。

[0079] 逻辑结构:每台物理机上运维节点为必须运行节点,主要负责加入系统云后对整个云系统的维护、数据同步、信息判断、仲裁等等。而分配节点、工作节点、工作程序在物理机上不一定必须运行,运行模式为在运维节点进行信息同步后决定启动哪些分配节点和工作节点,以及分配节点和工作节点之间如何形成归属关系,包括工作节点应该启动哪些工

作程序。

[0080] 在线升级:采用failover/failback,升级时部分节点先升级,然后逐步全部升级,实现不停机不停服无感升级。

[0081] 逻辑节点定义:

[0082] 运维节点:运维节点主要承担任务均衡的工作,包括分配节点、工作节点、其他节点加入云后的归属关系均衡、客户初次加入云后的链接信息分配等。运维节点间为环状组织结构,数据信息在环上为顺序流动,同正向或同逆向。各个运维节点的重要性是平等的,没有主备之分,这是与其他同类集群框架的重要差异性之一。

[0083] 分配节点:分配节点主要承担直接与客户端通信,接收来自客户端的任务,并将一个完整的业务均衡到不同的工作节点上,追踪任务直到完成,达到提高任务完成速度的目的,也就是要具备任务分割和转发功能。分配节点的实现逻辑不能涉及具体的业务,同时应该具备全息性,单分配节点也能完成多分配节点能完成的任务。与客户端直接通信还应该具备接入身份验证、任务数限制功能。在控制指令方面原则上接收来自运维节点的指令,推送指令到客户端;而信息同步是推送信息到运维节点,接收工作节点的信息。

[0084] 工作节点:一台物理设备上运行一个工作节点,工作节点接收其所属分配节点的数据,映射任务到下属的工作程序,同时维护下属工作程序的负载情况并反馈给分配节点。工作节点需要能兼容下属工作程序的并发性,包括无限制并发、有限制并发和串行。另外工作节点需要维护一张描述工作程序属性的表,用于汇总工作节点下属工作程序的负载情况,并同时同步到所属的分配节点,作为分配节点负载均衡的判断依据。

[0085] 工作程序:工作程序对应一种或一类具体的业务,每个工作程序在分布式框架中对应一个具体的业务编号,作为进程独立存在。例如负责位置解析的工作程序只完成位置解析任务,负责视频分析的工作程序只完成视频分析任务。工作程序的通信协议必须遵从现有的分布式框架通信协议。

[0086] 客户端程序:客户端是用户直接调用的程序,一般以动态库等形式存在,为调用者提供具体的调用接口。客户端通常需要提供并发调用、接入合法性验证、简单负载均衡功能。

[0087] 实施例2

[0088] 如图2所示,本实施例与实施例1的区别在于,包括上述的计算服务集群框架,还包括由自定义模块层、框架模块层和网络层组成的逻辑结构;

[0089] 所述网络层包括网络库C++接口、网络库Python接口、网络库C接口和Python通用库中的至少一个;

[0090] 所述框架模块层包括运维节点模块、分配节点模块和工作节点模块,还包括与运维节点模块、分配节点模块和工作节点模块均连接的数据结构模块、线程函数模块、过程模块、协议模块和Util模块;

[0091] 数据结构模块:负责自定义数据结构和数据处理;

[0092] 线程函数模块:负责实现各种线程函数,包括例测线程、任务线程、网络函数回调线程等;

[0093] 过程模块:负责实现各种过程,包括源语映射等;

[0094] 协议模块:负责控制协议和任务协议等各种协议数据的打包和解包工作;

[0095] Util模块:包括实现其他功能的函数群。

[0096] 自定义模块层:用户根据实际业务需求案子框架协议定义的功能。

[0097] 所述自定义模块层包括工作程序模块和客户端模块。

[0098] 网络层:主要负责提供整个框架所需的网络组建接口,包括多种语言接口;

[0099] 框架模块层:构成框架必备模块,分为运维节点模块、分配节点模块和工作节点模块,上述三个模块还包括多个功能模块,功能模块包括数据结构模块、线程函数模块、过程模块、协议模块和Util模块,

[0100] 实施例3

[0101] 如图3-5所示,基于分布式的计算服务集群系统的任务处理方法,包括上述的计算服务集群框架,还包括一个装有与计算服务集群框架匹配的客户端程序的客户端,还包括以下步骤:

[0102] S1、客户端向集群维护者申请调用账号、密码;客户端程序通过账号、密码向运维节点发起登录请求;

[0103] S2、运维节点验证客户的程序发送的账号、密码,验证成功后运维节点获取分配节点信息,返回分配信息到客户端;

[0104] S3、客户端根据运维节点返回的分配信息链接分配节点;

[0105] S4、客户端发送任务请求到分配节点,分配节点根据实际的任务类型和下属的工作节点负载情况综合配置任务请求的分割方式和接受任务的工作节点,然后将任务转发到接手任务的工作节点;

[0106] S5、接收到任务的工作节点调取对应的工作程序,将任务转发到工作程序进行处理,工作程序完成任务后将任务结果返回到工作节点;

[0107] S6、工作节点将任务结果返回分配该任务的分配节点;

[0108] S7、接收到任务结果的分配节点将任务结果返回到发送任务请求的客户端。

[0109] 分布式集群通常由n台物理设备组合而成,每台物理设备中部署全套共4种二进制程序,参照前面介绍的框架拓扑结构来部署。对外提供1个或多个集群入口地址,启动了分配节点的物理节点即为一个入口地址,整套系统中可以有多个分配节点被启动。

[0110] 客户端调用者需要向集群维护者申请调用账号密码,通过账号密码首先向集群入口地址发起访问,由入口运维节点返回实际访问地址,这一步对调用者不可见。

[0111] 认证通过后即可发起任务请求,客户端的任务请求到达分配节点后,由分配节点根据实际的任务类型和下属的工作节点负载情况综合裁定任务请求的分割方式和接受任务的工作节点。

[0112] 工作节点接收到任务请求后根据请求类型和下属工作程序的负载情况决定接受任务的工作程序。工作程序在完成任务后根据请求类型按原来返回结果或者不返回。

[0113] 在整个数据处理流程中各个节点都会根据任务请求类型做出对应的响应,包括是否要等待任务完成、是否是按并发处理或串行处理等等,都是有一套完整的通信协议来控制的,本发明技术关键点不在通信协议,所以对通信协议不做详细阐述。

[0114] 为了维护整个集群框架的稳定正常运转,各个逻辑节点间的相互控制关系也是极其重要的。

[0115] 首先运维节点是重中之重,每个物理设备必须运行一个运维节点,由多个运维节

点构成信息环,运维节点间的数据在环上顺序流动。运维节点主动向分配节点、工作节点推送控制指令,而分配节点、工作节点有信息同步需求则要主动推送到运维节点。

[0116] 实施例4

[0117] 本实施例与实施例3的区别在于,所述步骤S5中,接收到任务的工作节点为任务进行编号,并将编号和发送任务的分配节点匹配后进行记录,在步骤S6中,工作节点通过读取任务结果的任务编号,通过任务编号匹配发送任务的分配节点,将任务结果返回该分配节点。

[0118] 进一步的,所述运维节点在工作时包括主动过程任务和被动过程任务。

[0119] 主动过程任务为当前阶段主动向向量的前后阶段发送请求获取相应的信息;

[0120] 被动过程任务为当前节点接收到相邻节点的请求信息后作出相应的信息反馈。

[0121] 进一步的,所述主动过程任务和被动过程任务均包括加入云请求、退出云请求、相邻节点健康检查、表信息收集、表信息同步、新增节点、删除节点和获取信息表中的至少一个。

[0122] 进一步的,所述各个运维节点的功能相同,每个运维节点加入系统后在本地保存一张全表信息,包含系统上所有运维节点的状态等信息,在节点被删除或出现异常时其他节点的全表信息会得到更新,每个节点通过例测线程检测前后相连节点的健康状态并定期做全表信息同步,信息表中多节点在环中的位置是按加入顺序生成的Hashid固定的。

[0123] 基于前面对运维节点的功能介绍,可见框架对异常情况的处理就主要集中在运维节点中,而异常情况的处理合理性直接决定了整个框架的健壮性。

[0124] 关键点就在于运维节点的等效性,即是各个运维节点的功能相同,不存在常规分布式框架中的仲裁等机制,多运维节点环上的任何节点丢失都不影响系统的正常运行,具体保障机制如下所述:

[0125] 云管理过程:每个运维节点加入云后在本地保存一张全表信息,包含云上所有运维节点的状态等信息,在节点被删除或出现异常时其他节点的全表信息会得到更新,每个节点通过例测线程检测前后相连节点的健康状态并定期做全表信息同步,信息表中多节点在环中的位置是按加入顺序生成的Hashid固定的。详情可参见例图5。

[0126] 云均衡过程:每个运维节点负责控制本机工作节点/分配节点的启动和关闭,监测本机分配节点/工作节点的工作状态,通过一系列主动例测线程实现这一功能。

[0127] 通过上述保障机制在运维节点环中能够实现动态调整节点,达到整个环状结构中只要有一个节点正常运行,则整个系统就能正常对外提供服务,从而有效提升了系统的健壮性。

[0128] 实施例5

[0129] 本实施例为本系统一种典型工作平台,采用x86或x64平台、Windows或Linux系统。最大支持1000台物理设备相连,每台物理设备最少256M可用内存。

[0130] 实施例6

[0131] 如图4所示,本实施例为实施例3的一种具体使用实例,10062为客户的连接分配节点的端口号;10063为运维节点用于用户认证的端口号;30330:多物理设备间运维节点的连接端口,用于构成Hash环框架;30333:运维几点主动连接分配节点的端口号;30335运维节点主动连接工作节点的端口号;30336:分配节点主动连接工作节点的端口号;31004:工作

程序的监听端口号,用于工作程序的主动连接;10062、10063、31004端口用户可以进行自主配置。

[0132] 以上所述,仅为本发明的优选实施方式,但本发明的保护范围并不局限于此,任何熟悉本领域的技术人员在本发明所揭露的技术范围内,可不经创造性劳动想到的变化或替换,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该以权利要求书所限定的保护范围为准。

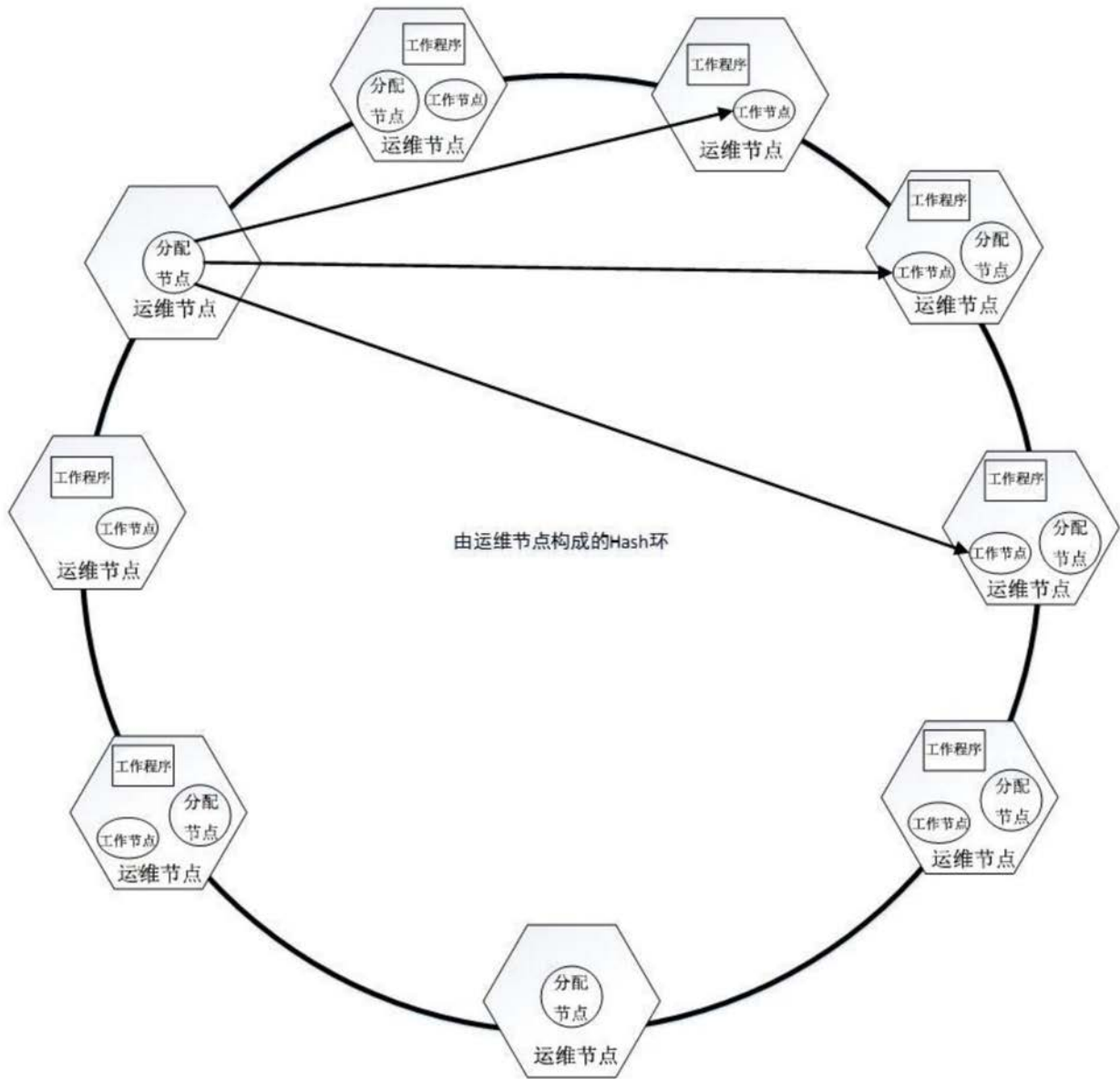


图1

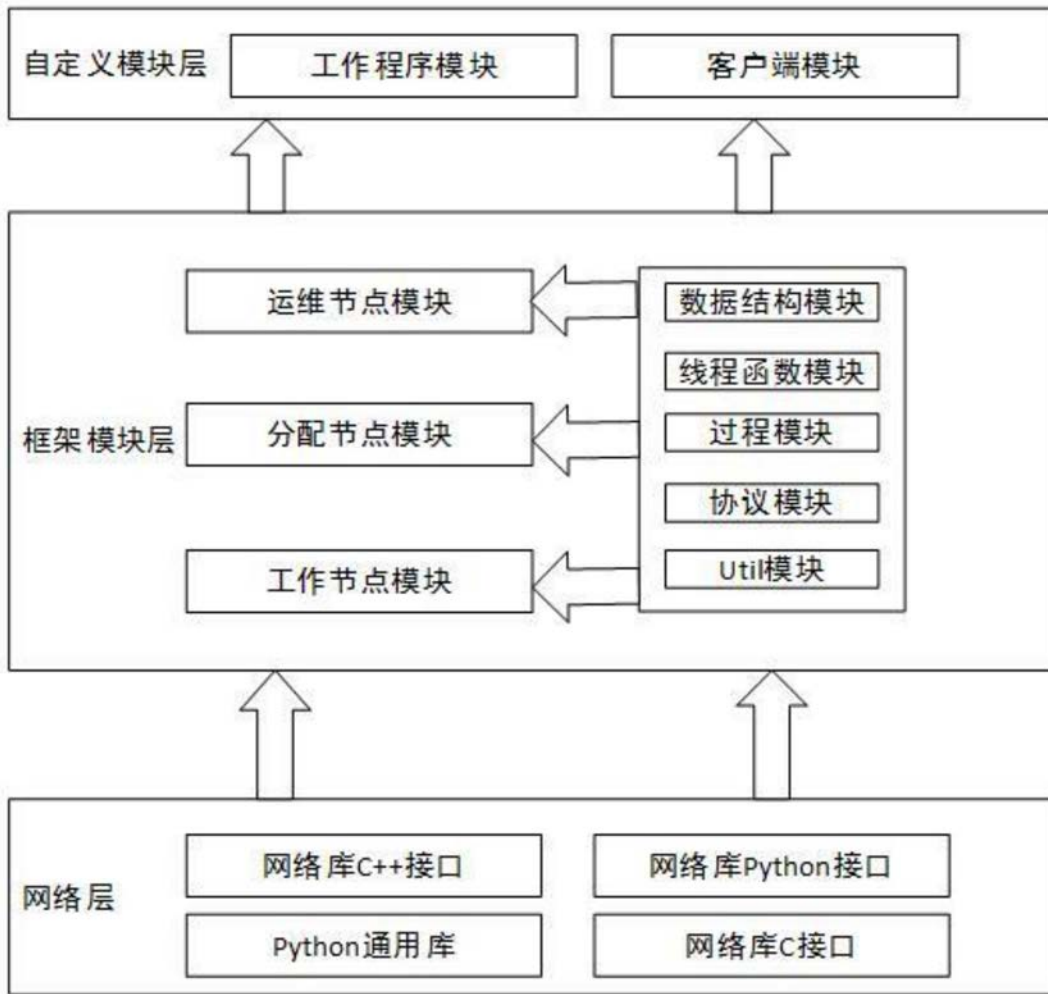


图2

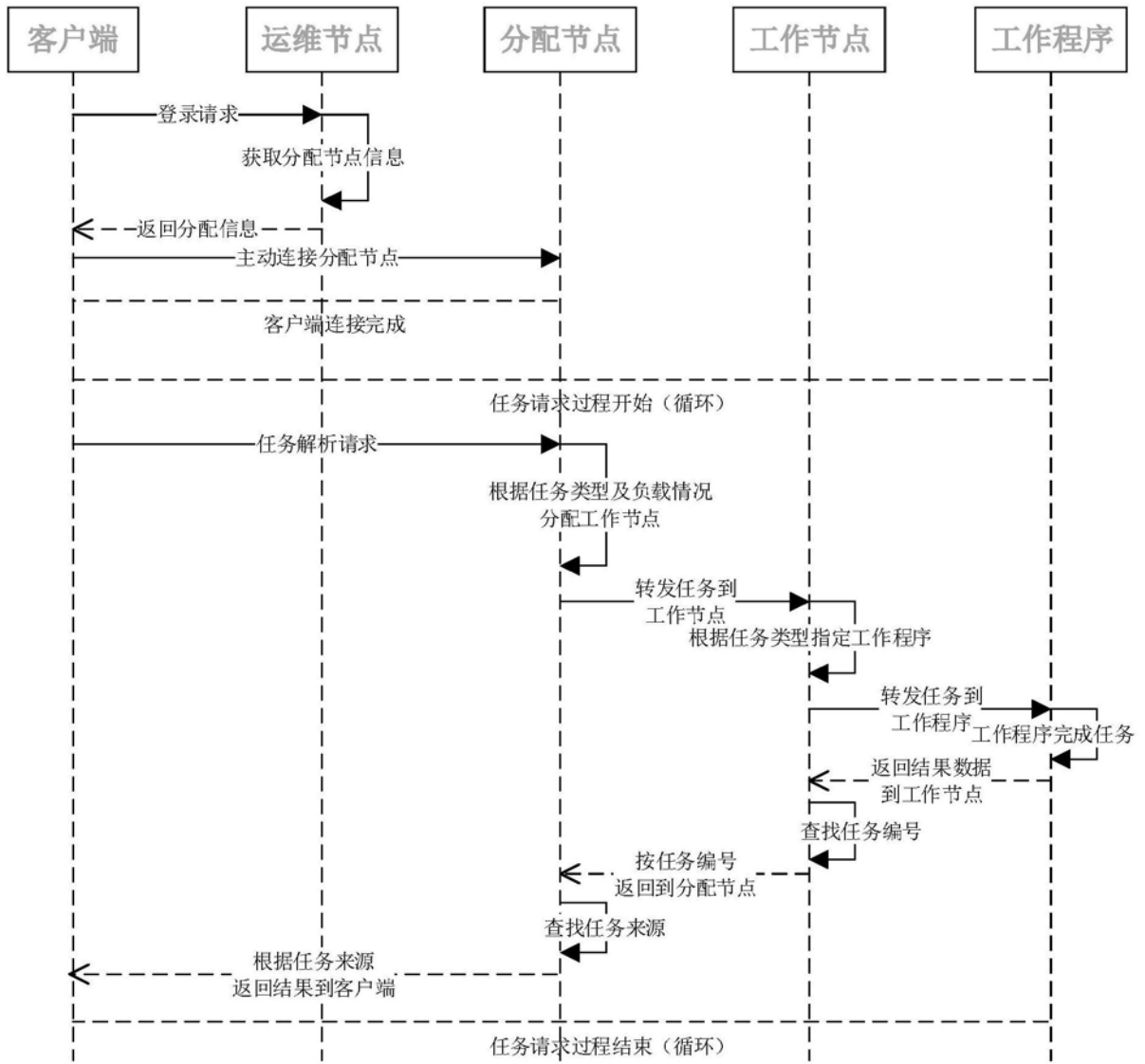


图3

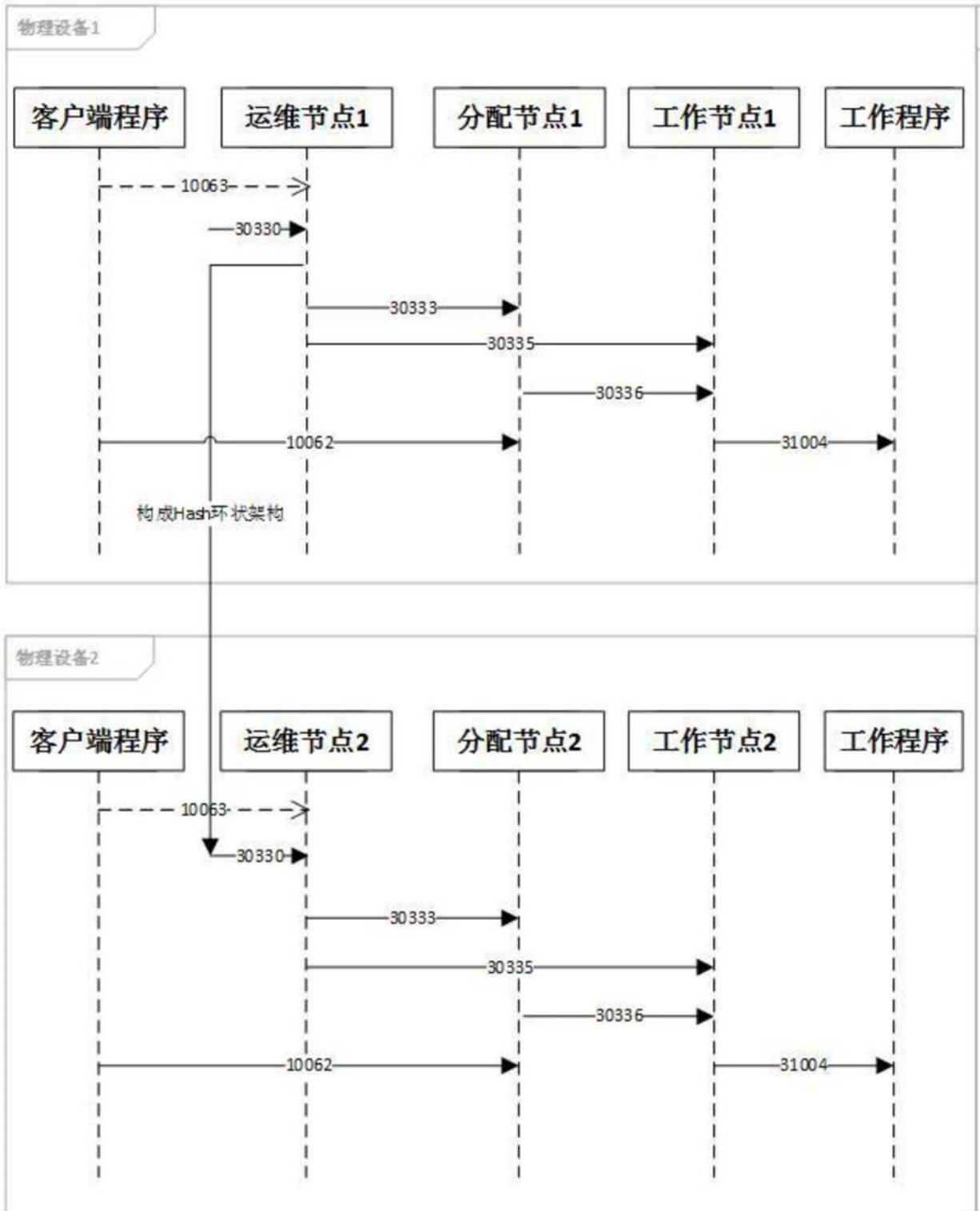


图4

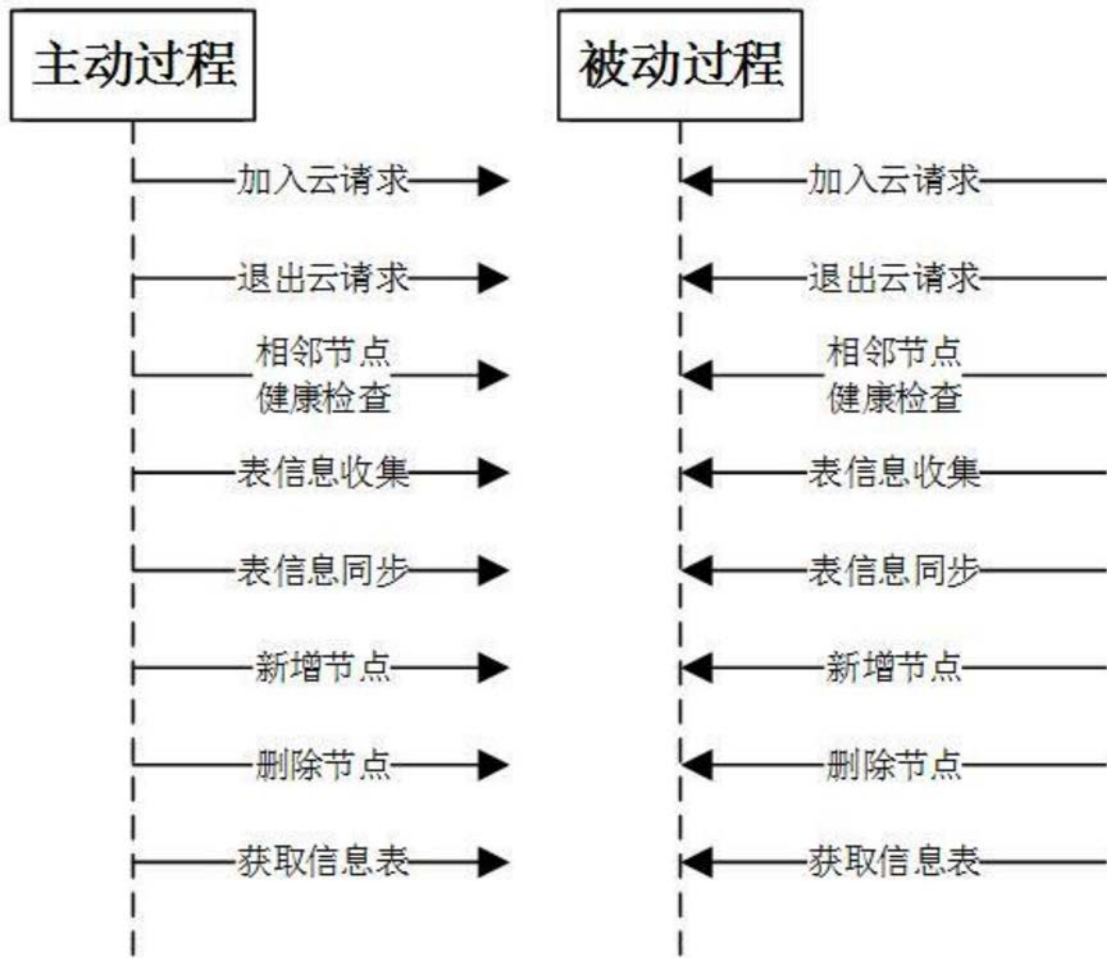


图5