



(12) 发明专利申请

(10) 申请公布号 CN 111880939 A

(43) 申请公布日 2020.11.03

(21) 申请号 202010793060.9

(22) 申请日 2020.08.07

(71) 申请人 曙光信息产业(北京)有限公司  
地址 100000 北京市海淀区东北旺西路8号  
院36号楼

申请人 曙光信息产业股份有限公司  
中科曙光国际信息产业有限公司

(72) 发明人 余彬 吕灼恒 张晋锋 李斌

(74) 专利代理机构 北京超凡宏宇专利代理事务  
所(特殊普通合伙) 11463

代理人 蒋姗

(51) Int. Cl.

G06F 9/50 (2006.01)

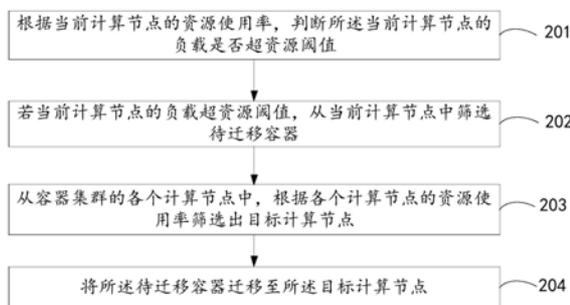
权利要求书2页 说明书12页 附图3页

(54) 发明名称

容器动态迁移方法、装置及电子设备

(57) 摘要

本申请提供了一种容器动态迁移方法、装置及电子设备,其中,该方法包括:根据当前计算节点的资源使用率,判断当前计算节点的负载是否超资源阈值;若当前计算节点的负载超资源阈值,从当前计算节点中筛选待迁移容器;从容器集群的各个计算节点中,根据各个计算节点的资源使用率筛选出目标计算节点;将待迁移容器迁移至目标计算节点。能够实现容器的迁移,从而实现容器集群负载均衡。



1. 一种容器动态迁移方法,其特征在于,包括:

根据当前计算节点的资源使用率,判断所述当前计算节点的负载是否超资源阈值;

若当前计算节点的负载超资源阈值,从当前计算节点中筛选待迁移容器;

从容器集群的各个计算节点中,根据各个计算节点的资源使用率筛选出目标计算节点;

将所述待迁移容器迁移至所述目标计算节点。

2. 根据权利要求1所述的方法,其特征在于,所述从容器集群的各个计算节点中,根据各个计算阶段的资源使用率筛选出目标计算节点,包括:

根据所述容器集群的各个计算节点的资源使用率,对各个计算节点按照从资源使用率低至高进行排序,得到计算节点排序表;

从所述计算节点排序表中选择排在前第一数量的计算节点,作为初始计算节点组;

从所述初始计算节点组中按照设定概率算法选出目标计算节点。

3. 根据权利要求2所述的方法,其特征在于,所述从所述初始计算节点组中按照设定概率算法选出目标计算节点,包括:

根据所述初始计算节点组中的各个计算节点的资源负载数据,确定出各个计算节点的下发概率,其中,所述初始计算节点组中的各个计算节点的下发概率之和为1;

随机确定一个小于1的随机正数;

根据所述随机正数以及各个计算节点的下发概率确定出目标计算节点。

4. 根据权利要求3所述的方法,其特征在于,所述根据所述初始计算节点组中的各个计算节点的资源负载数据,确定出各个计算节点的下发概率,通过以下公式实现:

$$P_i = (1 - \frac{R_i}{\sum_{i=1}^e R_i}) / (e - 1);$$

其中, $P_i$ 表示第*i*个计算节点的下发概率, $R_i$ 表示第*i*个计算节点对应的资源使用率, $e$ 表示初始计算节点组中的计算节点的数量。

5. 根据权利要求1-4任意一项所述的方法,其特征在于,所述从当前计算节点中筛选待迁移容器,包括:

根据所述当前计算节点的资源使用率确定出负载超资源阈值的资源类型;

根据所述资源类型从当前计算节点中选出待迁移容器。

6. 根据权利要求5所述的方法,其特征在于,所述根据所述资源类型从当前计算节点中选出待迁移容器,包括:

根据所述资源类型从当前计算节点中选出第一初始待迁移容器组;

从所述第一初始待迁移容器组中选出执行任务所需时长小于时长阈值的待迁移容器。

7. 根据权利要求5所述的方法,其特征在于,所述根据所述资源类型从当前计算节点中选出待迁移容器,包括:

根据所述资源类型从当前计算节点中选出第二初始待迁移容器组;

针对所述第二初始待迁移容器组的目标容器,确定所述目标容器对应的目标服务,以及所述目标服务对应的所属容器数量,所述目标容器为所述第二初始待迁移容器组中的任意一容器;

若所述所属容器数量大于数量阈值,则将所述目标容器作为待迁移容器。

8. 根据权利要求1所述的方法,其特征在于,所述根据当前计算节点的资源使用率,判断所述当前计算节点的负载是否超资源阈值,包括:

按照设定的时间规律获取当前计算节点的资源使用率,判断连续第二数量时间节点的资源使用率是否超资源阈值;

其中,若连续第二数量时间节点的资源使用率超资源阈值,则判定当前计算节点的负载超资源阈值。

9. 一种容器动态迁移装置,其特征在于,包括:

判断模块,用于根据当前计算节点的资源使用率,判断所述当前计算节点的负载是否超资源阈值;

第一筛选模块,用于若当前计算节点的负载超资源阈值,从当前计算节点中筛选待迁移容器;

第二筛选模块,用于从容器集群的各个计算节点中,根据各个计算节点的资源使用率筛选出目标计算节点;

迁移模块,用于将所述待迁移容器迁移至所述目标计算节点。

10. 一种电子设备,其特征在于,包括:处理器、存储器,所述存储器存储有所述处理器可执行的机器可读指令,当电子设备运行时,所述机器可读指令被所述处理器执行时执行如权利要求1至8任一所述的方法的步骤。

## 容器动态迁移方法、装置及电子设备

### 技术领域

[0001] 本申请涉及计算机技术领域,具体而言,涉及一种容器动态迁移方法、装置及电子设备。

### 背景技术

[0002] 随着云服务规模的扩大,数据中心不断地扩张,为了可以在大规模的云服务平台下有效实现负载均衡,需要将计算资源进行有效分配。

[0003] 在容器集群中,当大量任务同时运行和调度时,则可能会出现资源抢占、负载过度的异常情况;当容器集群规模特别大时,由于各种任务的大量频繁运行,同一个节点部署过多容器时,会使得节点资源使用率过大,导致集群负载不均衡,从而导致计算节点出现故障。对于这种情况,目前现有技术中的操作方式是通过集群管理工具发觉异常,并在异常恢复之前不再对异常计算节点下发任务,没有对异常的计算节点进行主动恢复。

### 发明内容

[0004] 本申请的目的在于提供一种容器动态迁移方法、装置及电子设备,能够实现容器的迁移,从而实现容器集群负载均衡。

[0005] 第一方面,本申请实施例提供一种容器动态迁移方法,包括:

[0006] 根据当前计算节点的资源使用率,判断所述当前计算节点的负载是否超资源阈值;

[0007] 若当前计算节点的负载超资源阈值,从当前计算节点中筛选待迁移容器;

[0008] 从容器集群的各个计算节点中,根据各个计算节点的资源使用率筛选出目标计算节点;

[0009] 将所述待迁移容器迁移至所述目标计算节点。

[0010] 在一可选的实施方式中,所述从容器集群的各个计算节点中,根据各个计算阶段的资源使用率筛选出目标计算节点,包括:

[0011] 根据所述容器集群的各个计算节点的资源使用率,对各个计算节点按照资源使用率从低至高进行排序,得到计算节点排序表;

[0012] 从所述计算节点排序表中选择排在前第一数量的计算节点,作为初始计算节点组;

[0013] 从所述初始计算节点组中按照设定概率算法选出目标计算节点。

[0014] 在本申请实施例中,通过排序的方式,选出当前负载轻的几个计算节点中选出其中对应的目标计算节点,而不是直接选择负载最轻的计算节点,从而可以减少因大量容器被迁移至同一计算节点而导致抖动现象,从而能提高迁移过程中的稳定性。其中,抖动现象为计算节点的资源使用率突然上升至一较高的值。

[0015] 在一可选的实施方式中,所述从所述初始计算节点组中按照设定概率算法选出目标计算节点,包括:

[0016] 根据所述初始计算节点组中的各个计算节点的资源负载数据,确定出各个计算节点的下发概率,其中,所述初始计算节点组中的各个计算节点的下发概率之和为1;

[0017] 随机确定一个小于1的随机正数;

[0018] 根据所述随机正数以及各个计算节点的下发概率确定出目标计算节点。

[0019] 在一可选的实施方式中,所述根据所述初始计算节点组中的各个计算节点的资源负载数据,确定出各个计算节点的下发概率,通过以下公式实现:

$$[0020] \quad P_i = (1 - \frac{R_i}{\sum_{i=1}^e R_i}) / (e - 1)$$

[0021] 其中, $P_i$ 表示第*i*个计算节点的下发概率, $R_i$ 表示第*i*个计算节点对应的资源使用率, $e$ 表示初始计算节点组中的计算节点的数量。

[0022] 在本申请实施例中,通过随机正数结合各个计算节点的下发概率确定出对应的目标计算节点,可以使负载较轻的几个计算节点被选中是随机的,从而可以减少因大量容器被迁移至同一计算节点而导致抖动现象。

[0023] 在一可选的实施方式中,所述从当前计算节点中筛选待迁移容器,包括:

[0024] 根据所述当前计算节点的资源使用率确定出负载超资源阈值的资源类型;

[0025] 根据所述资源类型从当前计算节点中选出待迁移容器。

[0026] 在本申请实施例中,通过确定出负载超资源阈值的资源类型,从而可以从使用资源类型的容器中选出待迁移容器,从而使迁移后的当前计算节点的负载更均衡。

[0027] 在一可选的实施方式中,所述根据所述资源类型从当前计算节点中选出待迁移容器,包括:

[0028] 根据所述资源类型从当前计算节点中选出第一初始待迁移容器组;

[0029] 从所述第一初始待迁移容器组中选出执行任务所需时长小于时长阈值的待迁移容器。

[0030] 在本申请实施例中,通过迁移所需时长较短的容器进行迁移,从而可以降低迁移代价,从而使当前计算节点能够更稳定的运行。

[0031] 在一可选的实施方式中,所述根据所述资源类型从当前计算节点中选出待迁移容器,包括:

[0032] 根据所述资源类型从当前计算节点中选出第二初始待迁移容器组;

[0033] 针对所述第二初始待迁移容器组的目标容器,确定所述目标容器对应的目标服务,以及所述目标服务对应的所属容器数量,所述目标容器为所述第二初始待迁移容器组中的任一容器;

[0034] 若所述所属容器数量大于数量阈值,则将所述目标容器作为待迁移容器。

[0035] 在本申请实施例中,通过选择某一服务中所属容器数量较多的服务中的容器作为待迁移的容器,从而可以降低容器的迁移对一计算节点中的工作的影响。

[0036] 在一可选的实施方式中,所述根据当前计算节点的资源使用率,判断所述当前计算节点的负载是否超资源阈值,包括:

[0037] 按照设定的时间规律获取当前计算节点的资源使用率,判断连续第二数量时间节点的资源使用率是否超资源阈值;

[0038] 其中,若连续第二数量时间节点的资源使用率超资源阈值,则判定当前计算节点

的负载超资源阈值。

[0039] 在本申请实施例中,通过几个时间点的判断确定出可以进行容器迁移,从而减少因为计算节点的资源使用率出现正常的抖动现象时,导致的误判,从而可以提高确定异常计算节点的准确率。

[0040] 第二方面,本申请实施例提供一种容器动态迁移装置,包括:

[0041] 判断模块,用于根据当前计算节点的资源使用率,判断所述当前计算节点的负载是否超资源阈值;

[0042] 第一筛选模块,用于若当前计算节点的负载超资源阈值,从当前计算节点中筛选待迁移容器;

[0043] 第二筛选模块,用于从容器集群的各个计算节点中,根据各个计算节点的资源使用率筛选出目标计算节点;

[0044] 迁移模块,用于将所述待迁移容器迁移至所述目标计算节点。

[0045] 第三方面,本申请实施例提供一种电子设备,包括:处理器、存储器,所述存储器存储有所述处理器可执行的机器可读指令,当电子设备运行时,所述机器可读指令被所述处理器执行时执行上述的方法的步骤。

[0046] 本申请实施例的有益效果是:通过选出资源使用率高的计算节点的容器进行迁移,从而可以降低计算节点的负载。

## 附图说明

[0047] 为了更清楚地说明本申请实施例的技术方案,下面将对实施例中所需要使用的附图作简单地介绍,应当理解,以下附图仅示出了本申请的某些实施例,因此不应被看作是对范围的限定,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相关的附图。

[0048] 图1为本申请实施例提供的电子设备的方框示意图。

[0049] 图2为本申请实施例提供的容器动态迁移方法的流程图。

[0050] 图3为本申请实施例提供的容器动态迁移方法的步骤202的详细流程图。

[0051] 图4为本申请实施例提供的容器动态迁移方法的当前计算节点中的容器分布示意图。

[0052] 图5为本申请实施例提供的容器动态迁移方法的步骤203的详细流程图。

[0053] 图6为本申请实施例提供的容器动态迁移装置的功能模块示意图。

## 具体实施方式

[0054] 下面将结合本申请实施例中附图,对本申请实施例中的技术方案进行描述。

[0055] 应注意到:相似的标号和字母在下面的附图中表示类似项,因此,一旦某一项在一个附图中被定义,则在随后的附图中不需要对其进行进一步定义和解释。同时,在本申请的描述中,术语“第一”、“第二”等仅用于区分描述,而不能理解为指示或暗示相对重要性。

[0056] 首先,先对本申请实施例会使用到的概念进行介绍。

[0057] Docker是一种开源的应用容器引擎,让开发者可以打包他们的应用以及依赖包到一个可移植的镜像中,然后发布到任何流行的Linux或Windows机器上,也可以实现虚拟化。

容器是完全使用沙箱机制,相互之间不会有任何接口。

[0058] Swarm是Docker官方开发的原生集群工具,提供了资源调度、负载均衡、服务发现等功能。非常轻量并易于使用,为开发者提供了高纬度的配置方式,能够快速实现具体的工作流。

[0059] Kubernetes是Google开源的容器集群管理系统,其提供应用部署、维护、扩展机制等功能,利用Kubernetes能方便地管理跨机器运行容器化的应用。主要实现语言为Go语言。其中,Kubernetes缩写为K8s。

[0060] Mesos是Apache旗下的开源分布式资源管理框架。

[0061] 负载均衡是指将负载(工作任务)进行平衡、分摊到多个操作单元上进行运行,例如FTP服务器、Web服务器、企业核心应用服务器和其它主要任务服务器等,从而协同完成工作任务。

[0062] 当Docker集群规模特别大时,由于各种任务的大量频繁运行,同一个计算节点部署过多容器时,会使得该计算节点资源使用率过大,导致集群负载不均衡。目前面对异常恢复的场景,无论是Swarm、Kubernetes或Mesos都未行成有效的解决方案。常见的处理方式是停止为对应的计算节点分配任务,没有对异常的计算节点进行主动恢复。

[0063] 基于上述研究,本申请实施例提供一种容器动态迁移方法、装置及电子设备,通过对异常计算节点中的容器进行迁移,从而实现集群的负载均衡。下面通过几个实施例对本申请进行介绍。

[0064] 实施例一

[0065] 为便于对本实施例进行理解,首先对执行本申请实施例所公开的容器动态迁移方法的电子设备进行详细介绍。

[0066] 如图1所示,是电子设备的方框示意图。电子设备100可以包括存储器111、存储控制器112、处理器113、外设接口114。本领域普通技术人员可以理解,图1所示的结构仅为示意,其并不对电子设备100的结构造成限定。例如,电子设备100还可包括比图1中所示更多或者更少的组件,或者具有与图1所示不同的配置。

[0067] 上述的存储器111、存储控制器112、处理器113、外设接口114各元件相互之间直接或间接地电性连接,以实现数据的传输或交互。例如,这些元件相互之间可通过一条或多条通讯总线或信号线实现电性连接。上述的处理器113用于执行存储器中存储的可执行模块。

[0068] 其中,存储器111可以是,但不限于,随机存取存储器(Random Access Memory,简称RAM),只读存储器(Read Only Memory,简称ROM),可编程只读存储器(Programmable Read-Only Memory,简称PROM),可擦除只读存储器(Erasable Programmable Read-Only Memory,简称EPROM),电可擦除只读存储器(Electric Erasable Programmable Read-Only Memory,简称EEPROM)等。其中,存储器111用于存储程序,所述处理器113在接收到执行指令后,执行所述程序,本申请实施例任一实施例揭示的过程定义的电子设备的100所执行的方法可以应用于处理器113中,或者由处理器113实现。

[0069] 上述的处理器113可能是一种集成电路芯片,具有信号的处理能力。上述的处理器113可以是通用处理器,包括中央处理器(Central Processing Unit,简称CPU)、网络处理器(Network Processor,简称NP)等;还可以是数字信号处理器(digital signal processor,简称DSP)、专用集成电路(Application Specific Integrated Circuit,简称

ASIC)、现场可编程门阵列(FPGA)或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件。可以实现或者执行本申请实施例中的公开的各方法、步骤及逻辑框图。通用处理器可以是微处理器或者该处理器也可以是任何常规的处理器等。

[0070] 上述的外设接口114将各种输入/输出装置耦合至处理器113以及存储器111。在一些实施例中,外设接口114,处理器113以及存储控制器112可以在单个芯片中实现。在其他一些实例中,他们可以分别由独立的芯片实现。

[0071] 本实施例中的电子设备100可以用于执行本申请实施例提供的各个方法中的各个步骤。下面通过几个实施例详细描述容器动态迁移方法的实现过程。

[0072] 实施例二

[0073] 请参阅图2,是本申请实施例提供的容器动态迁移方法的流程图。下面将对图2所示的具体流程进行详细阐述。

[0074] 步骤201,根据当前计算节点的资源使用率,判断所述当前计算节点的负载是否超资源阈值。

[0075] 可选地,计算节点的资源可以包括CPU(Central Processing Unit,中央处理器)、GPU(Graphics Processing Unit,图形处理器)、内存等资源。

[0076] 本实施例中,可以通过计算节点的资源确定出计算节点的负载情况。

[0077] 可选地,每一类型的资源分别对应有不同的资源阈值。

[0078] 示例性地,针对CPU设置的资源阈值为CPU使用率阈值 $C_{max}$ ,针对GPU设置的资源阈值为GPU使用率阈值 $G_{max}$ ,针对内存设置的资源阈值为内存使用率阈值 $N_{max}$ 。示例性地,不同的计算节点的处理能力可能不同,因此针对不同的计算节点设置的资源阈值也可以不同。

[0079] 可选地,以CPU、GPU、内存为例,任何一种类型的资源使用率超过对应的资源阈值时,则可以判定当前计算节点负载超资源阈值。

[0080] 进一步地,考虑计算节点的资源使用率可能会出现正常的抖动现象,为了提高判断的准确率,可以对当前计算节点进行周期性的监控。通过多个周期的监控,判断当前计算节点是否超负载。

[0081] 示例性地,步骤201可以包括:按照设定的时间规律获取当前计算节点的资源使用率,判断连续第二数量时间节点的资源使用率是否超资源阈值。

[0082] 其中,若连续第二数量时间节点的资源使用率超资源阈值,则判定当前计算节点的负载超资源阈值。

[0083] 可选地,上述的第二数量的取值可以按照需求设置。例如,第二数量的取值可以是4、5、6等值。

[0084] 示例性地,可以每隔十分钟获取一次当前计算节点的资源使用率,且将获得的资源使用率与预设的资源阈值进行比较,分别确定当前计算节点的各个时间节点的资源使用率是否超资源阈值。

[0085] 可选地,上述的设定的时间规律对应的时间间隔可以按照需求设置,例如,时间间隔也可以是五分钟、二十分钟、半小时等。

[0086] 以CPU,以及第二数量的取值为4为例,若连续四次获得的当前计算节点的CPU使用率都超过CPU使用率阈值 $C_{max}$ ,则可以判定当前计算节点的CPU使用超负载。则需要对当前

计算节点的容器进行迁移,以减轻当前计算节点的CPU的负载。

[0087] 步骤202,若当前计算节点的负载超资源阈值,从当前计算节点中筛选待迁移容器。

[0088] 示例性地,筛选的待迁移容器可以是一个或多个容器。

[0089] 在一种实施方式中,如图3所示,步骤202可以包括以下步骤。

[0090] 步骤2021,根据所述当前计算节点的资源使用率确定出负载超资源阈值的资源类型。

[0091] 例如,若当前计算节点的CPU的使用率超资源阈值,则负载超资源阈值的资源类型为CPU。

[0092] 步骤2022,根据所述资源类型从当前计算节点中选出待迁移容器。

[0093] 可选地,选出的待迁移容器的数量可以是一个,也可以是多个。

[0094] 示例性地,若当前计算节点的CPU使用率超资源阈值,则可以将CPU需求较高的一个或多个容器作为待迁移容器。

[0095] 示例性地,若当前计算节点的GPU使用率超资源阈值,则可以将GPU需求较高的一个或多个容器作为待迁移容器。

[0096] 可选地,每次迁移的容器数量可以为多个,当多项资源超资源阈值时,则可以基于多项超资源阈值的资源使用情况选择多个容器进行迁移。当仅一项资源超资源阈值时,则可以基于超资源阈值的一项资源使用情况选择多个容器进行迁移。

[0097] 示例性地,若当前计算节点的GPU使用率和内存使用率均超资源阈值,则可以将GPU需求较高和内存需求较高的容器作为待迁移容器。例如,可以从GPU需求较高的容器中选出第一容器,从内存需求较高的容器中选出第二容器,将该第一容器和第二容器作为待迁移容器。

[0098] 示例性地,若当前计算节点的GPU使用率和内存使用率均超资源阈值,则可以将GPU需求较高和内存需求较高的容器作为待迁移容器。例如,可以从GPU需求较高的容器中选出第一容器组,从内存需求较高的容器中选出第二容器组。可选地,可以将该第一容器组和第二容器组作为待迁移容器。可选地,也可以从该第一容器组和第二容器组选出多个容器作为待迁移容器。

[0099] 示例性地,若当前计算节点的内存使用率超资源阈值,则可以将内存需求较高的容器作为待迁移容器。例如,可以从内存需求较高的容器中选出多个容器。可选地,可以将选出的多个容器作为待迁移容器。

[0100] 可选地,每次迁移的容器数量也可以仅为一个,当多项资源超资源阈值时,则可以基于多项超资源阈值的资源使用情况选择一个容器进行迁移。当仅一项资源超资源阈值时,则可以基于超资源阈值的一项资源使用情况选择一个容器进行迁移。

[0101] 示例性地,若当前计算节点的GPU使用率和内存使用率均超资源阈值,则可以将GPU需求较高或内存需求较高的容器作为待迁移容器。例如,可以从GPU需求较高的容器中选出第三容器,从内存需求较高的容器中选出第四容器,将该第三容器或第四容器作为待迁移容器。

[0102] 示例性地,若当前计算节点的CPU使用率超资源阈值,则可以将CPU需求较高的容器作为待迁移容器。例如,可以从CPU需求较高的容器中选出一个容器。可选地,可以将选出

的容器作为待迁移容器。

[0103] 考虑各个计算节点中的容器可能均存在运行任务,容器的迁移可能导致运行任务被搁置,因此,为了降低容器的迁移影响正在进行的任务,则可以从执行任务时所需执行时间更短的容器中进行筛选。

[0104] 在第一种实施方式中,步骤2022可以被实施为:根据所述资源类型从当前计算节点中选出第一初始待迁移容器组;从所述第一初始待迁移容器组中选出执行任务所需时长小于时长阈值的待迁移容器。

[0105] 可选地,若当前计算节点的CPU使用率超资源阈值,则可以先从当前计算节点中选出CPU使用率最高的一个或多个容器作为第一初始待迁移容器组。

[0106] 示例性地,可以先对当前计算节点中的容器的CPU使用率从高到低进行排序,将排序在前N位的容器选为第一初始待迁移容器组,其中N为正整数。

[0107] 可选地,若当前计算节点的CPU使用率和内存使用率超资源阈值,则可以先从当前计算节点中选出CPU使用率最高的一个或多个容器,以及内存使用率最高的一个或多个容器作为第一初始待迁移容器组。

[0108] 可选地,若当前计算节点的CPU使用率和内存使用率超资源阈值,则可以先从当前计算节点中选出CPU使用率最高的一个或多个容器,或者从当前计算节点中选出内存使用率最高的一个或多个容器作为第一初始待迁移容器组。

[0109] 可选地,若当前计算节点的CPU使用率、内存使用率和GPU使用率均超资源阈值,则可以从当前计算节点中选出CPU使用率最高的一个或多个容器,或者从当前计算节点中选出内存使用率最高的一个或多个容器,或者从当前计算节点中选出GPU使用率最高的一个或多个容器作为第一初始待迁移容器组。

[0110] 可选地,若当前计算节点的CPU使用率、内存使用率和GPU使用率均超资源阈值,也可以基于CPU使用率、GPU使用率和内存使用率选出CPU使用率较高的一个或多个容器,和GPU使用率较高的一个或多个容器,和内存使用率较高的一个或多个容器作为第一初始待迁移容器组。

[0111] 其中,第一初始迁移容器组的数量可以是一个,也可以是多个。

[0112] 示例性地,若当前的第一初始待迁移容器组中仅包括一个容器,且该容器的执行任务所需时长大于时长阈值,则不对当前的计算节点进行容器迁移。

[0113] 示例性地,若当前的第一初始待迁移容器组包括多个容器,且该第一初始待迁移容器的各个容器存在的执行任务所需时长小于时长阈值,则将所有执行任务所需时长小于时长阈值的容器作为待迁移容器。

[0114] 示例性地,若当前的第一初始待迁移容器组包括多个容器,该第一初始待迁移容器的各个容器不存在的执行任务所需时长小于时长阈值,则不对当前的计算节点进行容器迁移。

[0115] 可选地,可以提前设置能够被选为待迁移容器的执行任务所需时长的上限。示例性地,时长阈值可以按照需求设置。例如,不同的计算节点设置的时长阈值可以相同,也可以不同。

[0116] 例如,设置的时长阈值为15min时,则被选出的待迁移容器执行任务所需时长小于该15min。

[0117] 再例如,设置的时长阈值为10min时,则被选出的待迁移容器执行任务所需时长小于该10min。

[0118] 由于同时移动同一服务下的容器可能导致该服务被中断,因此,为了避免一服务因为容器的迁移而被中断,在选择待迁移容器时,则可以将一服务下的容器数量作为考虑因素。

[0119] 在第二种实施方式中,步骤2022可以被实施为:根据所述资源类型从当前计算节点中选出第二初始待迁移容器组;针对所述第二初始待迁移容器组的目标容器,确定所述目标容器对应的目标服务,以及所述目标服务对应的所属容器数量;若目标服务对应的所属容器数量大于数量阈值,则将所述目标容器作为待迁移容器。

[0120] 示例性地,若一容器为一服务的所属容器,则表示该容器是用于执行该服务的任务的容器。

[0121] 其中,上述的目标容器为该第二初始待迁移容器组中的任一容器。

[0122] 关于第二初始待迁移容器组的确定方式,可以与前述的第一初始待迁移容器组的确定方式相同,在此不再赘述。

[0123] 在一个实例中,如图4所示,当前计算节点中可以运行有三项服务,分别为:服务一、服务二和服务三。其中,用于执行服务一对应的任务的容器有C1、C2、C3、C4,则服务一的所属容器数量为四。用于执行服务二对应的任务的容器有C5、C6、C7、C8、C9、C10、C11,则服务二的所属容器数量为七。用于执行服务三对应的任务的容器有C12、C13、C14、C15、C16,则服务三的所属容器数量为五。

[0124] 例如,在此实例中,第二初始待迁移容器组中包括容器C1、C8、C14。

[0125] 在一个实例中,设定的数量阈值可以是五,该第二初始待迁移容器组中的一容器对应的目标服务当前的容器数量为五或小于五,则该容器不能被选作待迁移容器。

[0126] 如图4所示,C1对应的服务为服务一,C8对应的服务为服务二,C14对应的服务为服务三。由于服务一和服务三对应的所属容器数量都小于或等于五,因此,在此实例中C1、C14不能被选作待迁移容器。由于服务二对应的所属容器数量大于五,因此,在此实例中C8可以被选作待迁移容器。

[0127] 在一个实例中,设定数量阈值可以是四,该第二初始待迁移容器组中的一容器对应的目标服务当前的容器数量为五,则该容器可以被选作待迁移容器。

[0128] 如图4所示,C1对应的服务为服务一,C8对应的服务为服务二,C14对应的服务为服务三。由于服务一对应的所属容器数量等于四,因此,在此实例中C1不能被选作待迁移容器。由于服务二和服务三对应的所属容器数量均大于五,因此,在此实例中C8和C14可以被选作待迁移容器。可选地,在此实例中C8和C14可以均被选为待迁移容器。可选地,也可以从C8和C14中随机选择一个容器作为待迁移容器。

[0129] 可选地,可以将一服务下的容器数量,以及容器执行任务所需时长均作为考虑因素。

[0130] 在第三种实施方式中,步骤2022可以被实施为:根据所述资源类型从当前计算节点中选出第三初始待迁移容器组;从该第三初始待迁移容器组中选出任务所需时长小于时长阈值的容器,且第三初始待迁移容器组中的容器对应的服务下的所属容器数量大于数量阈值的容器,作为待迁移容器。

[0131] 关于第三种实施方式中的其它细节可以参阅第一种实施方式和第二种实施方式中的描述,在此不再赘述。

[0132] 步骤203,从容器集群的各个计算节点中,根据各个计算节点的资源使用率筛选出目标计算节点。

[0133] 选择目标计算节点可以选择当前容器集群中性能最好、负载最轻的节点。但是当容器集群规模较大时,这种方法可能会出现剧烈抖动现象,即在某一时刻会将大量容器同时迁移到同一个目标计算节点时,从而会造成该目标计算节点的负载急剧上升,从而产生剧烈抖动现象。

[0134] 为了解决以上问题,如图5所示,步骤203可以包括以下步骤。

[0135] 步骤2031,根据所述容器集群的各个计算节点的资源使用率,对各个计算节点按照资源使用率从低至高进行排序,得到计算节点排序表。

[0136] 示例性地,若当前计算节点是因为CPU使用率超资源阈值,则可以将容器集群中的计算节点的CPU使用率进行排序。

[0137] 示例性地,若当前计算节点是因为GPU使用率超资源阈值,则可以将容器集群中的计算节点的GPU使用率进行排序。

[0138] 示例性地,若当前计算节点是因为内存使用率超资源阈值,则可以将容器集群中的计算节点的内存使用率进行排序。

[0139] 步骤2032,从所述计算节点排序表中选择排在前第一数量的计算节点,作为初始计算节点组。

[0140] 可选地,第一数量的取值可以按照需求设置。例如,第一数量的取值可以是3、5、7等值。

[0141] 可选地,容器集群的规模越大,第一数量的取值也可以越大。

[0142] 步骤2033,从所述初始计算节点组中按照设定概率算法选出目标计算节点。

[0143] 可选地,可以根据所述初始计算节点组中的各个计算节点的资源负载数据,确定出各个计算节点的下发概率;然后,随机确定一个小于1的随机正数;根据所述随机正数以及各个计算节点的下发概率确定出目标计算节点。

[0144] 其中,初始计算节点组中的各个计算节点的下发概率之和为1。

[0145] 可选地,根据所述初始计算节点组中的各个计算节点的资源负载数据,确定出各个计算节点的下发概率,通过以下公式实现:

$$[0146] \quad P_i = (1 - \frac{R_i}{\sum_{i=1}^e R_i}) / (e - 1);$$

[0147] 其中, $P_i$ 表示第*i*个计算节点的下发概率, $R_i$ 表示第*i*个计算节点对应的资源使用率, $e$ 表示初始计算节点组中的计算节点的数量。

[0148] 可选地,各个计算节点的下发概率也可以根据各个计算节点的资源使用率,设置各个计算节点的下发概率为等差数列的下发概率。例如,计算节点的资源使用率越低对应的下发概率越高。

[0149] 如果容器集群较大,且当前需要迁移的容器较多时,如果多个需要被迁移的容器被迁移至同一个计算节点中,则可能会导致该计算节点发生抖动现象。因此,通过使用上述的随机正数从负载较低的几个计算节点确定出目标计算节点,从而可以使容器被迁移至负

载较低的几个计算节点概率相同,从而减少为了满足迁移需求却导致计算节点抖动的问题的情况。

[0150] 示例性地,该根据所述随机正数以及各个计算节点的下发概率确定出目标计算节点包括:可以将随机正数与各个计算节点的下发概率作差,将与随机正数差值最小的计算节点作为目标计算节点。

[0151] 示例性地,该根据所述随机正数以及各个计算节点的下发概率确定出目标计算节点包括:根据各个计算节点的下发概率,为每个计算节点设置概率区间,该随机正数所在的概率区间对应的计算节点作为目标计算节点。

[0152] 步骤204,将所述待迁移容器迁移至所述目标计算节点。

[0153] 通过本申请实施例中的方法,可以更准确地定位到资源负载过高的计算节点,从而准确地筛选出可能出现异常的计算节点;能够准确地匹配到需要被迁移的容器;还能够相对准确定位到需要迁移的容器合适的迁移目标计算节点。进一步地,由于迁移至的目标计算节点采用的概率筛选的方式,能够达到均匀迁移的效果,防止迁移目标计算节点也出现负载过高的情况。进一步地,由于本申请实施例中的方案是通过容器的迁移的方式处理集群中的各个计算节点的负载不均衡的问题,因此不存在计算节点暂停服务(暂停下发任务)的情况,则不需要对计算节点进行恢复等操作,也能够实现各个计算节点的持续工作。

[0154] 实施例三

[0155] 基于同一申请构思,本申请实施例中还提供了与容器动态迁移方法对应的容器动态迁移装置,由于本申请实施例中的装置解决问题的原理与前述的容器动态迁移方法实施例相似,因此本实施例中的装置的实施可以参见上述方法的实施例中的描述,重复之处不再赘述。

[0156] 请参阅图6,是本申请实施例提供的容器动态迁移装置的功能模块示意图。本实施例中的容器动态迁移装置中的各个模块用于执行上述方法实施例中的各个步骤。容器动态迁移装置包括:判断模块301、第一筛选模块302、第二筛选模块303以及迁移模块304;其中,

[0157] 判断模块301,用于根据当前计算节点的资源使用率,判断所述当前计算节点的负载是否超资源阈值;

[0158] 第一筛选模块302,用于若当前计算节点的负载超资源阈值,从当前计算节点中筛选待迁移容器;

[0159] 第二筛选模块303,用于从容器集群的各个计算节点中,根据各个计算节点的资源使用率筛选出目标计算节点;

[0160] 迁移模块304,用于将所述待迁移容器迁移至所述目标计算节点。

[0161] 一种可能的实施方式中,第二筛选模块303,包括:得到单元、选择单元以及选出单元,其中,

[0162] 排序单元,用于根据所述容器集群的各个计算节点的资源使用率,对各个计算节点按照资源使用率从低至高进行排序,得到计算节点排序表;

[0163] 选择单元,用于从所述计算节点排序表中选择排在前第一数量的计算节点,作为初始计算节点组;

[0164] 选出单元,用于从所述初始计算节点组中按照设定概率算法选出目标计算节点。

[0165] 一种可能的实施方式中,选择单元,用于:

[0166] 根据所述初始计算节点组中的各个计算节点的资源负载数据,确定出各个计算节点的下发概率,其中,所述初始计算节点组中的各个计算节点的下发概率之和为1;

[0167] 随机确定一个小于1的随机正数;

[0168] 根据所述随机正数以及各个计算节点的下发概率确定出目标计算节点。

[0169] 一种可能的实施方式中,所述根据所述初始计算节点组中的各个计算节点的资源负载数据,确定出各个计算节点的下发概率,通过以下公式实现:

$$[0170] \quad P_i = (1 - \frac{R_i}{\sum_{i=1}^e R_i}) / (e - 1);$$

[0171] 其中, $P_i$ 表示第*i*个计算节点的下发概率, $R_i$ 表示第*i*个计算节点对应的资源使用率, $e$ 表示初始计算节点组中的计算节点的数量。

[0172] 一种可能的实施方式中,第一筛选模块302包括:类型确定单元和容器选出单元;

[0173] 类型确定单元,用于根据所述当前计算节点的资源使用率确定出负载超资源阈值的资源类型;

[0174] 容器选出单元,用于根据所述资源类型从当前计算节点中选出待迁移容器。

[0175] 一种可能的实施方式中,容器选出单元,用于:

[0176] 根据所述资源类型从当前计算节点中选出第一初始待迁移容器组;

[0177] 从所述第一初始待迁移容器组中选出执行任务所需时长小于时长阈值的待迁移容器。

[0178] 一种可能的实施方式中,容器选出单元,用于:

[0179] 根据所述资源类型从当前计算节点中选出第二初始待迁移容器组;

[0180] 针对所述第二初始待迁移容器组的目标容器,确定所述目标容器对应的目标服务,以及所述目标服务对应的所属容器数量,所述目标容器为所述第二初始待迁移容器组中的任一容器;

[0181] 若所述所属容器数量大于数量阈值,则将所述目标容器作为待迁移容器。

[0182] 一种可能的实施方式中,判断模块301,用于:

[0183] 按照设定的时间规律获取当前计算节点的资源使用率,判断连续第二数量时间节点的资源使用率是否超资源阈值;

[0184] 其中,若连续第二数量时间节点的资源使用率超资源阈值,则判定当前计算节点的负载超资源阈值。

[0185] 此外,本申请实施例还提供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,该计算机程序被处理器运行时执行上述方法实施例中所述的容器动态迁移方法的步骤。

[0186] 本申请实施例所提供的容器动态迁移方法的计算机程序产品,包括存储了程序代码的计算机可读存储介质,所述程序代码包括的指令可用于执行上述方法实施例中所述的容器动态迁移方法的步骤,具体可参见上述方法实施例,在此不再赘述。

[0187] 在本申请所提供的几个实施例中,应该理解到,所揭露的装置和方法,也可以通过其它的方式实现。以上所描述的装置实施例仅仅是示意性的,例如,附图中的流程图和框图显示了根据本申请的多个实施例的装置、方法和计算机程序产品的可能实现的体系架构、功能和操作。在这点上,流程图或框图中的每个方框可以代表一个模块、程序段或代码的一

部分,所述模块、程序段或代码的一部分包含一个或多个用于实现规定的逻辑功能的可执行指令。也应当注意,在有些作为替换的实现方式中,方框中所标注的功能也可以以不同于附图中所标注的顺序发生。例如,两个连续的方框实际上可以基本并行地执行,它们有时也可以按相反的顺序执行,这依所涉及的功能而定。也要注意的,框图和/或流程图中的每个方框、以及框图和/或流程图中的方框的组合,可以用执行规定的功能或动作的专用的基于硬件的系统来实现,或者可以用专用硬件与计算机指令的组合来实现。

[0188] 另外,在本申请各个实施例中的各功能模块可以集成在一起形成一个独立的部分,也可以是各个模块单独存在,也可以两个或两个以上模块集成形成一个独立的部分。

[0189] 所述功能如果以软件功能模块的形式实现并作为独立的产品销售或使用,可以存储在一个计算机可读取存储介质中。基于这样的理解,本申请的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的部分可以以软件产品的形式体现出来,该计算机软件产品存储在一个存储介质中,包括若干指令用以使得一台计算机设备(可以是个人计算机,服务器,或者网络设备)执行本申请各个实施例所述方法的全部或部分步骤。而前述的存储介质包括:U盘、移动硬盘、只读存储器(ROM,Read-Only Memory)、随机存取存储器(RAM,Random Access Memory)、磁碟或者光盘等各种可以存储程序代码的介质。需要说明的是,在本文中,诸如第一和第二等之类的关系术语仅仅用来将一个实体或者操作与另一个实体或操作区分开来,而不一定要求或者暗示这些实体或操作之间存在任何这种实际的关系或者顺序。而且,术语“包括”、“包含”或者任何其他变体意在涵盖非排他性的包含,从而使得包括一系列要素的过程、方法、物品或者设备不仅包括那些要素,而且还包括没有明确列出的其他要素,或者是还包括为这种过程、方法、物品或者设备所固有的要素。在没有更多限制的情况下,由语句“包括……”限定的要素,并不排除在包括所述要素的过程、方法、物品或者设备中还存在另外的相同要素。

[0190] 以上所述仅为本申请的优选实施例而已,并不用于限制本申请,对于本领域的技术人员来说,本申请可以有各种更改和变化。凡在本申请的精神和原则之内,所作的任何修改、等同替换、改进等,均应包含在本申请的保护范围之内。应注意到:相似的标号和字母在下面的附图中表示类似项,因此,一旦某一项在一个附图中被定义,则在随后的附图中不需要对其进行进一步定义和解释。

[0191] 以上所述,仅为本申请的具体实施方式,但本申请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申请揭露的技术范围内,可轻易想到变化或替换,都应涵盖在本申请的保护范围之内。因此,本申请的保护范围应以权利要求的保护范围为准。

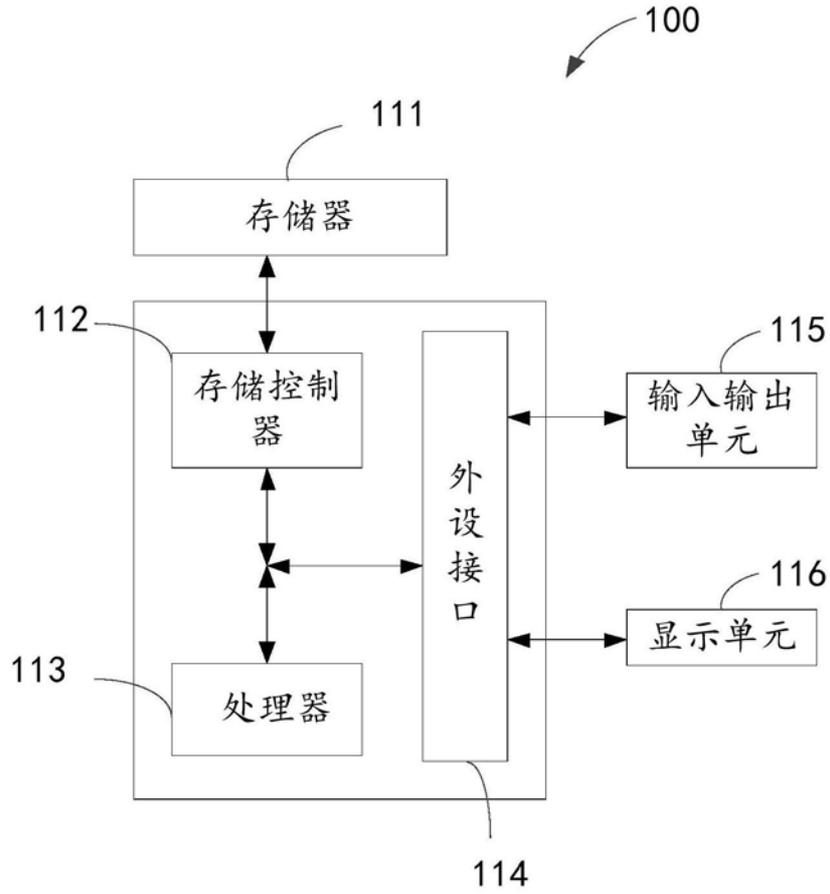


图1

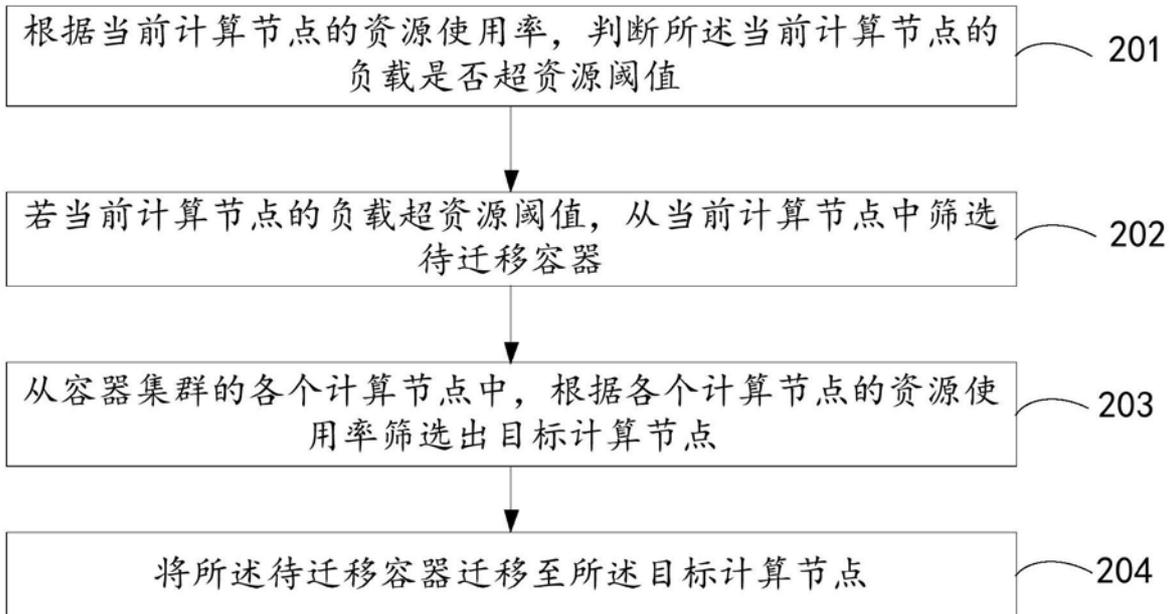


图2

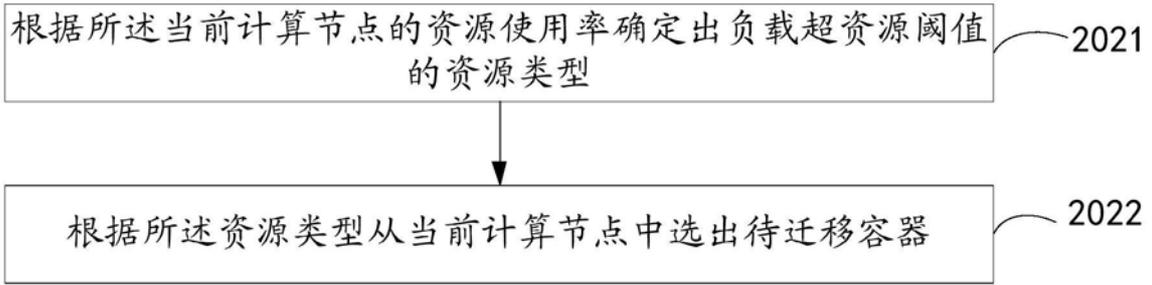


图3

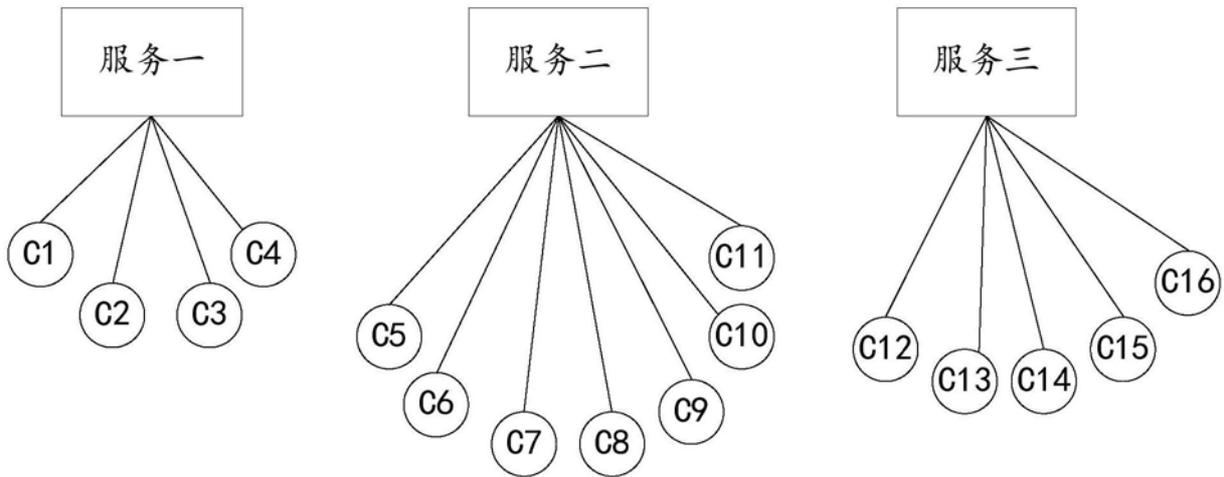


图4

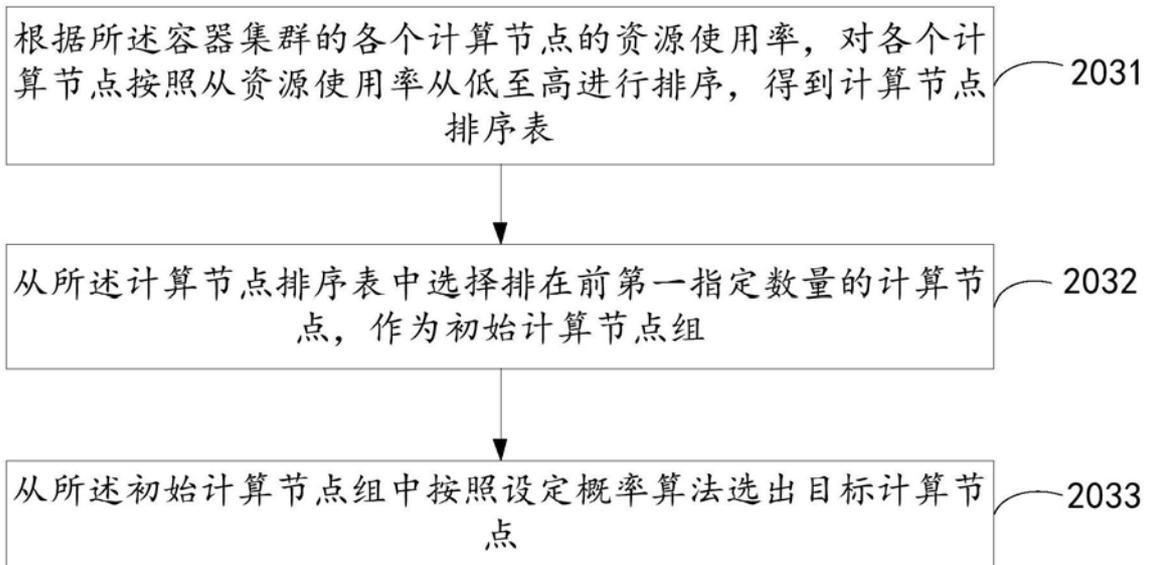


图5

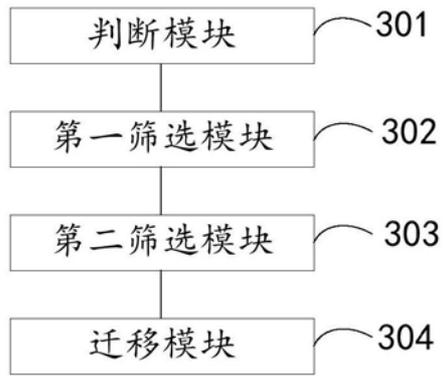


图6