



(12) 发明专利申请

(10) 申请公布号 CN 116708303 A

(43) 申请公布日 2023. 09. 05

(21) 申请号 202310905071.5

(22) 申请日 2023.07.21

(71) 申请人 济南浪潮数据技术有限公司

地址 250000 山东省济南市自由贸易试验区济南片区浪潮路1036号浪潮科技园S05楼S311室

(72) 发明人 李凤娟 朱志文

(74) 专利代理机构 北京三聚阳光知识产权代理有限公司 11250

专利代理师 熊飞雪

(51) Int. Cl.

H04L 47/125 (2022.01)

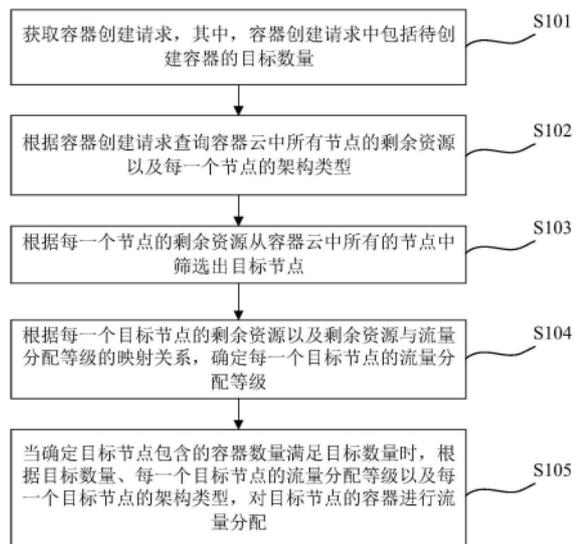
权利要求书2页 说明书12页 附图4页

(54) 发明名称

流量分配方法、装置、计算机设备及存储介质

(57) 摘要

本发明涉及计算机技术领域,公开了一种流量分配方法、装置、计算机设备及存储介质,包括:获取容器创建请求,根据容器创建请求查询容器云中所有节点的剩余资源以及每一个节点的架构类型;根据每一个节点的剩余资源从容器云中所有的节点中筛选出目标节点;根据每一个目标节点的剩余资源以及剩余资源与流量分配等级的映射关系,确定每一个目标节点的流量分配等级;当确定目标节点包含的容器数量满足目标数量时,根据目标数量、每一个目标节点的流量分配等级以及每一个目标节点的架构类型,对目标节点的容器进行流量分配。本发明可以基于不同架构进行流量的合理分配,可以让业务系统能够持续健壮稳定的提供服务。



1. 一种流量分配方法,其特征在于,所述方法包括:

获取容器创建请求,其中,所述容器创建请求中包括待创建容器的目标数量;

根据所述容器创建请求查询容器云中所有节点的剩余资源以及每一个所述节点的架构类型;

根据每一个所述节点的剩余资源从所述容器云中所有的节点中筛选出目标节点;

根据每一个所述目标节点的剩余资源以及剩余资源与流量分配等级的映射关系,确定每一个所述目标节点的流量分配等级;

当确定所述目标节点包含的容器数量满足所述目标数量时,根据所述目标数量、每一个所述目标节点的流量分配等级以及每一个目标节点的架构类型,对所述目标节点的容器进行流量分配。

2. 根据权利要求1所述的方法,其特征在于,所述剩余资源包括中央处理器的使用率,所述根据每一个所述节点的剩余资源从所述容器云中所有的节点中筛选出目标节点,包括:

当第一节点的中央处理器的使用率小于或者等于预设阈值时,确定所述第一节点为目标节点,其中,所述第一节点为所述容器云中任一个节点。

3. 根据权利要求1或2所述的方法,其特征在于,当确定所述目标节点包含的容器数量不满足所述目标数量时,所述方法还包括:

当所述容器数量大于所述目标数量时,将大于所述目标数量的容器进行驱逐,使所述目标节点的容器数量等于所述目标数量,

或者,

当所述容器数量小于所述目标数量时,在所述目标节点新建容器,使所述目标节点的容器数量等于所述目标数量。

4. 根据权利要求1或2所述的方法,其特征在于,所述流量分配等级包括至少一个,所述根据所述目标数量、每一个所述目标节点的流量分配等级以及每一个目标节点的架构类型,对所述目标节点的容器进行流量分配,包括:

从所有的流量分配等级中选取目标级别;

确定所述目标级别中属于第一架构类型的目标节点数量,其中,所述第一架构类型为所述目标级别对应的目标节点所属的架构类型中任一个架构类型;

根据所述目标数量,所述第一架构类型的目标节点数量,确定第一流量分配等级中所述第一架构类型的流量分配比例;

依次按照所述目标级别中每一种架构类型分别对应的流量分配比例,将待分配流量分配至所述目标级别分别对应的目标节点上;

当所述待分配流量仍有余量时,从流量分配等级中选取下一个流量分配等级,并利用下一个流量分配等级对应的目标节点对所述余量进行分配,直至所有的流量分配完毕为止。

5. 根据权利要求4所述的方法,其特征在于,所述从所有的流量分配等级中选取目标级别,包括:

选取流量分配等级中优先级最高的流量分配等级作为所述目标级别。

6. 根据权利要求4所述的方法,其特征在于,所述根据所述目标数量,所述第一架构类

型的目标节点数量,确定第一流量分配等级中所述第一架构类型的流量分配比例,具体通过如下公式确定:

$$L_1 = N_{pod} * N_{node1} * m1$$

其中, L_1 为所述第一架构类型的流量分配比例, N_{pod} 为所述目标数量, N_{node1} 为所述第一架构类型的目标节点数量, $m1$ 为所述第一架构类型的预设权重。

7. 根据权利要求1至3或5至6任一项所述的方法,其特征在于,所述在每一个所述目标节点创建所述目标数量的容器之后,所述方法还包括:

监测所述目标节点中每一个所述容器的状态;

当确定第一容器的状态为异常时,将所述第一容器从所述目标节点进行驱逐,并根据每一个所述目标节点的剩余容器数量、每一个所述目标节点的流量分配等级以及每一个目标节点的架构类型,重新对所述目标节点的容器进行流量分配,其中,所述第一容器为所述目标节点中任一个容器。

8. 一种流量分配装置,其特征在于,所述装置包括:

获取模块,用于获取容器创建请求,其中,所述容器创建请求中包括待创建容器的目标数量;

查询模块,用于根据所述容器创建请求查询容器云中所有节点的剩余资源以及每一个所述节点的架构类型;

筛选模块,用于根据每一个所述节点的剩余资源从所述容器云中所有的节点中筛选出目标节点;

确定模块,用于根据每一个所述目标节点的剩余资源以及剩余资源与流量分配等级的映射关系,确定每一个所述目标节点的流量分配等级;

分配模块,用于当确定所述目标节点包含的容器数量满足所述目标数量时,根据所述目标数量、每一个所述目标节点的流量分配等级以及每一个目标节点的架构类型,对所述目标节点的容器进行流量分配。

9. 一种计算机设备,其特征在于,包括:

存储器和处理器,所述存储器和所述处理器之间互相通信连接,所述存储器中存储有计算机指令,所述处理器通过执行所述计算机指令,从而执行权利要求1至7中任一项所述的流量分配方法。

10. 一种计算机可读存储介质,其特征在于,所述计算机可读存储介质上存储有计算机指令,所述计算机指令用于使计算机执行权利要求1至7中任一项所述的流量分配方法。

流量分配方法、装置、计算机设备及存储介质

技术领域

[0001] 本发明涉及计算机技术领域，具体涉及流量分配方法、装置、计算机设备及存储介质。

背景技术

[0002] 一云多芯是指用一套云操作系统来管理不同架构的硬件服务器集群。常规的云平台只能对一种架构的CPU进行硬件细节屏蔽，并通过云管理软件来管理，但是这种管理模式不能覆盖所有的用户需求，流量的分发只能基于同平台架构的容器，通过人为干预实现新旧版本的容器流量按分配比例进行流量的分发，新版本容器的环境与配置需要与旧版本保持一致，无形中增加了环境配置的压力，同时也增加了流量分发的时间和空间的局限性。流量的分发只能基于同一平台架构进行流量的引流，不能跨平台架构类型进行流量分发的自动调整，对环境的压力比较大，对环境稳定性也有很大挑战，而且存在资源浪费、运维复杂、无法统一管理、用户体验差等问题。

发明内容

[0003] 有鉴于此，本发明提供了一种流量分配方法、装置、计算机设备和存储介质，以解决不能跨平台架构类型进行流量分发的问题。

[0004] 第一方面，本发明提供了一种流量分配方法，包括：

[0005] 获取容器创建请求，其中，容器创建请求中包括待创建容器的目标数量；

[0006] 根据容器创建请求查询容器云中所有节点的剩余资源以及每一个节点的架构类型；

[0007] 根据每一个节点的剩余资源从容器云中所有的节点中筛选出目标节点；

[0008] 根据每一个目标节点的剩余资源以及剩余资源与流量分配等级的映射关系，确定每一个目标节点的流量分配等级；

[0009] 当确定目标节点包含的容器数量满足目标数量时，根据目标数量、每一个目标节点的流量分配等级以及每一个目标节点的架构类型，对目标节点的容器进行流量分配。

[0010] 本发明提供一种流量分配方法，具有如下优点：

[0011] 获取容器创建请求，根据容器创建请求查询容器云中所有节点的剩余资源以及每一个节点的架构类型；根据每一个节点的剩余资源从容器云中所有的节点中筛选出目标节点；根据每一个目标节点的剩余资源以及剩余资源与流量分配等级的映射关系，确定每一个目标节点的流量分配等级；当确定目标节点包含的容器数量满足目标数量时，根据目标数量、每一个目标节点的流量分配等级以及每一个目标节点的架构类型，对目标节点的容器进行流量分配。可以基于容器创建请求查询符合要求的节点作为流量分配的目标节点，然后根据架构类型，目标节点的剩余资源等，对流量进行分配，可以对流量以不同的维度进行灵活的分配，将请求分发到不同架构的不同容器上，实现负载均衡，在流量的分发上更高效，更能缓解各节点的业务压力，使系统能提供更稳定的运行环境，让业务系统能够持续健

壮稳定的提供服务。

[0012] 在一种可选的实施方式中,剩余资源包括中央处理器的使用率,根据每一个节点的剩余资源从容器云中所有的节点中筛选出目标节点,包括:

[0013] 当第一节点的中央处理器的使用率小于或者等于预设阈值时,确定第一节点为目标节点,其中,第一节点为容器云中任一个节点。

[0014] 在一种可选的实施方式中,当确定目标节点包含的容器数量不满足目标数量时,方法还包括:

[0015] 当容器数量大于目标数量时,将大于目标数量的容器进行驱逐,使目标节点的容器数量等于目标数量;

[0016] 或者,

[0017] 当容器数量小于目标数量时,在目标节点新建容器,使目标节点的容器数量等于目标数量。

[0018] 在一种可选的实施方式中,流量分配等级包括至少一个,根据目标数量、每一个目标节点的流量分配等级以及每一个目标节点的架构类型,对目标节点的容器进行流量分配,包括:

[0019] 从所有的流量分配等级中选取目标级别;

[0020] 确定目标级别中属于第一架构类型的目标节点数量,其中,第一架构类型为目标级别对应的目标节点所属的架构类型中任一个架构类型;

[0021] 根据目标数量,第一架构类型的目标节点数量,确定第一流量分配等级中第一架构类型的流量分配比例;

[0022] 依次按照目标级别中每一种架构类型分别对应的流量分配比例,将待分配流量分配至目标级别分别对应的目标节点上;

[0023] 当待分配流量仍有余量时,从流量分配等级中选取下一个流量分配等级,并利用下一个流量分配等级对应的目标节点对余量进行分配,直至所有的流量分配完毕为止。

[0024] 通过上述方式,可以将目标节点根据流量分配等级划分为多个目标级别,分别统计每个目标节点内的每一个架构类型的数量和每一个架构类型下目标节点的数量,确定每一个流量分配等级的流量分配比例,当目标级别未能够把所有的待分配流量分配完成时,再选下一个流量分配等级进行分配,可以灵活有序的对完成对流量的分配,平衡各个节点的压力,保证系统的稳定运行。

[0025] 在一种可选的实施方式中,从所有的流量分配等级中选取目标级别,包括:

[0026] 选取流量分配等级中优先级最高的流量分配等级作为目标级别。

[0027] 在一种可选的实施方式中,根据目标数量,第一架构类型的目标节点数量,确定第一流量分配等级中第一架构类型的流量分配比例,具体通过如下公式确定:

$$[0028] \quad L_1 = N_{pod} * N_{node1} * m1$$

[0029] 其中, L_1 为第一架构类型的流量分配比例, N_{pod} 为目标数量, N_{node1} 为第一架构类型的目标节点数量, $m1$ 为第一架构类型的预设权重。

[0030] 在一种可选的实施方式中,在每一个目标节点创建目标数量的容器之后,方法还包括:

[0031] 监测目标节点中每一个容器的状态;

[0032] 当确定第一容器的状态为异常时,将第一容器从目标节点进行驱逐,并根据每一个目标节点的剩余容器数量、每一个目标节点的流量分配等级以及每一个目标节点的架构类型,重新对目标节点的容器进行流量分配,其中,第一容器为目标节点中任一容器。

[0033] 第二方面,本发明提供了一种流量分配装置,包括:

[0034] 获取模块,用于获取容器创建请求,其中,容器创建请求中包括待创建容器的目标数量;

[0035] 查询模块,用于根据容器创建请求查询容器云中所有节点的剩余资源以及每一个节点的架构类型;

[0036] 筛选模块,用于根据每一个节点的剩余资源从容器云中所有的节点中筛选出目标节点;

[0037] 确定模块,用于根据每一个目标节点的剩余资源以及剩余资源与流量分配等级的映射关系,确定每一个目标节点的流量分配等级;

[0038] 分配模块,用于当确定目标节点包含的容器数量满足目标数量时,根据目标数量、每一个目标节点的流量分配等级以及每一个目标节点的架构类型,对目标节点的容器进行流量分配。

[0039] 第三方面,本发明提供了一种计算机设备,包括:存储器和处理器,存储器和处理器之间互相通信连接,存储器中存储有计算机指令,处理器通过执行计算机指令,从而执行上述第一方面或其对应的任一实施方式的流量分配方法。

[0040] 第四方面,本发明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指令,计算机指令用于使计算机执行上述第一方面或其对应的任一实施方式的流量分配方法。

附图说明

[0041] 为了更清楚地说明本发明具体实施方式或现有技术中的技术方案,下面将对具体实施方式或现有技术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方式,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。

[0042] 图1是根据本发明实施例的流量分配方法的流程示意图;

[0043] 图2是根据本发明实施例的另一流量分配方法的流程示意图;

[0044] 图3是根据本发明实施例的又一流量分配方法的流程示意图;

[0045] 图4是根据本发明实施例的又一流量分配方法的流程示意图

[0046] 图5是根据本发明实施例的流量分配装置的结构框图;

[0047] 图6是本发明实施例的计算机设备的硬件结构示意图。

具体实施方式

[0048] 为使本发明实施例的目的、技术方案和优点更加清楚,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。

[0049] 在具有不同架构的容器云平台中,一般使用云管软件来实现对不同架构类型的节点下的容器进行管理,但是这种方法并不能覆盖所有的客户需求,而且仍然存在资源浪费、运维复杂、无法统一维护管理、用户体验差等问题。而相关技术中的容器管理平台(Kubernetes,简称K8s)应用,流量的分发只能基于同平台架构的容器(pod),通过人为干预实现新旧版本的pod流量按分配比例进行流量的分发,新版本pod的环境与配置与旧版本保持一致,无形中增加了环境配置的压力,同时也增加了流量分发的时间和空间的局限性。

[0050] 基于此,根据本发明实施例,提供了一种流量分配方法实施例,需要说明的是,在附图的流程图示出的步骤可以在诸如一组计算机可执行指令的计算机系统中执行,并且,虽然在流程图中示出了逻辑顺序,但是在某些情况下,可以以不同于此处的顺序执行所示出或描述的步骤。

[0051] 在本实施例中提供了一种流量分配方法,可用于上述的计算机设备,如服务器、容器云平台等,图1是根据本发明实施例的流量分配方法的流程图,如图1所示,该流程包括如下步骤:

[0052] 步骤S101,获取容器创建请求。

[0053] 具体的,容器创建请求中包括待创建容器的目标数量。

[0054] 在一个可选的例子中,例如用户根据实际业务需要发起容器创建请求,请求中包含业务需要的容器的数量即目标数量。

[0055] 步骤S102,根据容器创建请求查询容器云中所有节点的剩余资源以及每一个节点的架构类型。

[0056] 具体的,当获取到创建请求后,查询容器云平台中所有节点的剩余资源以及每一个节点的架构类型,剩余资源指尚未使用的可以用于其他计算的资源。

[0057] 在一个可选的例子中,架构类型例如可以包括:超威半导体(Advanced Micro Devices,AMD)架构类型、进阶精简指令集机器(Advanced RISC Machine,ARM)架构类型等容器云平台可以包含的架构类型。

[0058] 步骤S103,根据每一个节点的剩余资源从容器云中所有的节点中筛选出目标节点。

[0059] 具体的,可以设定剩余资源阈值,当节点的剩余资源大于或者等于剩余资源阈值时,将该节点作为目标节点,遍历所有的节点,就可以筛选出所有的目标节点,目标节点为流量分配的候选节点。

[0060] 步骤S104,根据每一个目标节点的剩余资源以及剩余资源与流量分配等级的映射关系,确定每一个目标节点的流量分配等级。

[0061] 具体的,可以建立剩余资源与流量分配等级的映射关系,并将每一个目标节点的剩余资源与映射关系中的每一个流量分配等级进行匹配,确定目标节点的流量分配等级。

[0062] 在一个可选的例子中,例如目标节点A的剩余资源位于第一流量分配等级的区间内,则确定A的流量分配等级为第一流量分配等级,第一流量分配等级为所有的流量分配等级中任一流量分配等级。

[0063] 步骤S105,当确定目标节点包含的容器数量满足目标数量时,根据目标数量、每一个目标节点的流量分配等级以及每一个目标节点的架构类型,对目标节点的容器进行流量分配。

[0064] 具体的,当确定目标节点包含的容器数量满足目标数量时,可以根据流量分配等级确定目标节点的分配顺序,进而根据目标数量和目标节点的架构类型确定该目标节点在所属的流量分配等级下的分配策略,对该目标节点进行流量分配。

[0065] 在一个可选的例子中,可以将流量分配等级按一定的规则设置标号,按标号的顺序进行流量分配,当第一个标号的流量分配等级中的节点全部饱和时,再进行下一个标号的流量分配等级的流量分配操作。

[0066] 本实施例提供的流量分配方法,获取容器创建请求,根据容器创建请求查询容器云中所有节点的剩余资源以及每一个节点的架构类型;根据每一个节点的剩余资源从容器云中所有的节点中筛选出目标节点;根据每一个目标节点的剩余资源以及剩余资源与流量分配等级的映射关系,确定每一个目标节点的流量分配等级;当确定目标节点包含的容器数量满足目标数量时,根据目标数量、每一个目标节点的流量分配等级以及每一个目标节点的架构类型,对目标节点的容器进行流量分配。可以基于容器创建请求查询符合要求的节点作为流量分配的目标节点,然后根据架构类型,目标节点的剩余资源等,对流量进行分配,可以对流量以不同的维度进行灵活的分配,将请求分发到不同架构的不同容器上,实现负载均衡,在流量的分发上更高效,更能缓解各节点的业务压力,使系统能提供更稳定的运行环境,让业务系统能够持续健壮稳定的提供服务。

[0067] 在一种可选的实施方式中,剩余资源包括中央处理器的使用率,根据每一个节点的剩余资源从容器云中所有的节点中筛选出目标节点,包括:

[0068] 当第一节点的中央处理器的使用率小于或者等于预设阈值时,确定第一节点为目标节点,其中,第一节点为容器云中任一个节点。

[0069] 具体的,剩余资源包括中央处理器的使用率,例如CPU使用率,将CPU使用率小于或者等于预设阈值的节点作为目标节点,例如CPU使用率小于20%的节点,具体根据实际情况而定即可。

[0070] 在一种可选的实施方式中,当确定目标节点包含的容器数量不满足目标数量时,方法还包括:

[0071] 当容器数量大于目标数量时,将大于目标数量的容器进行驱逐,使目标节点的容器数量等于目标数量;

[0072] 或者,

[0073] 当容器数量小于目标数量时,在目标节点新建容器,使目标节点的容器数量等于目标数量。

[0074] 具体的,当确定目标节点后,获取目标节点下的容器数量,当目标节点下的容器数量与目标数量不一致时,对目标节点下的容器进行处理。当容器数量大于目标数量时,对多余的容器进行驱逐,保证该目标节点下的容器数量等于目标数量,当容器数量小于目标数量时,在目标节点创建新的容器,保证该目标节点下的容器数量等于目标数量。

[0075] 在一个可选的例子中,例如目标数量为5个,当目标节点A下的容器为8个时,驱逐3个容器到其他节点,使目标节点A下的容器数量为5个;当目标节点B下的容器数量为2个时,在目标节点B下新建3个容器,使目标节点B下的容器数量为5个。

[0076] 在本实施例中提供了一种流量分配方法,可用于上述的计算机设备,如服务器、容器云平台等,图2是根据本发明实施例的流量分配方法的流程图,如图2所示,该流程包括如

下步骤：

[0077] 步骤S201,获取容器创建请求。

[0078] 步骤S202,根据容器创建请求查询容器云中所有节点的剩余资源以及每一个节点的架构类型。

[0079] 步骤S203,根据每一个节点的剩余资源从容器云中所有的节点中筛选出目标节点。

[0080] 步骤S204,根据每一个目标节点的剩余资源以及剩余资源与流量分配等级的映射关系,确定每一个目标节点的流量分配等级。

[0081] 步骤S201、步骤S202、步骤S203以及步骤S204详细请分别参见图1所示实施例的步骤S101、步骤S102、步骤S103以及步骤S104,在此不再赘述。

[0082] 步骤S205,当确定目标节点包含的容器数量满足目标数量时,根据目标数量、每一个目标节点的流量分配等级以及每一个目标节点的架构类型,对目标节点的容器进行流量分配。

[0083] 其中,上述步骤S205包括:

[0084] 步骤S2051,从所有的流量分配等级中选取目标级别。

[0085] 具体的,可以为流量分配等级设定调度顺序,按照调度顺序,从所有的流量分配等级中选取目标级别。

[0086] 在一个可选的例子中,例如流量分配等级中CPU使用率低的可以优先调度,那么,可以从所有的流量分配等级中选择CPU使用率最低的流量分配等级作为目标级别。

[0087] 在一个可选的例子中,或者可以根据待分配流量的大小,从所有的流量分配等级中选择与待分配的流量大小最匹配的流量分配等级作为目标级别。

[0088] 步骤S2052,确定目标级别中属于第一架构类型的目标节点数量。

[0089] 具体的,第一架构类型为目标级别对应的目标节点所属的架构类型中任一架构类型。即确定目标级别中每一个架构类型的目标节点数量。例如,可以通过每一个节点的节点标识等信息,确定每一个节点的架构类型,然后统计每一个架构类型的目标节点数量。

[0090] 在一个可选的例子中,例如目标级别中包含ARM架构类型的节点5个,ADM架构类型的节点3个,其他架构类型的节点1个。

[0091] 步骤S2053,根据目标数量,第一架构类型的目标节点数量,确定第一流量分配等级中第一架构类型的流量分配比例。

[0092] 具体的,在一个可选的例子中,例如可以从选定的第一个目标级别开始进行流量调度,确定该目标级别中每一个架构类型的目标节点数量,确定该架构类型的流量分配比例,例如目标节点数量多的所分配的比例更高。

[0093] 在一种可选的实施方式中,根据目标数量,第一架构类型的目标节点数量,确定第一流量分配等级中第一架构类型的流量分配比例,具体通过如下公式确定:

[0094] $L_1 = N_{pod} * N_{node1} * m1$ (公式1)

[0095] 其中, L_1 为第一架构类型的流量分配比例, N_{pod} 为目标数量, N_{node1} 为第一架构类型的目标节点数量, $m1$ 为第一架构类型的预设权重。

[0096] 通过上述方式,可以结合实际情况,例如架构类型的属性等,给每一种架构类型设定不同的预设权重,可以确定目标级别内每一个架构类型的流量分配比例,并且实用性更

高。

[0097] 步骤S2054,依次按照目标级别中每一种架构类型分别对应的流量分配比例,将待分配流量分配至目标级别分别对应的目标节点上。

[0098] 具体的,在一个可选的例子中,例如待分配流量为50G,有三种架构类型,通过公式1的评估,第一种架构类型的流量分配比例为40%,第二种架构类型的流量分配比例为40%,第三种架构类型的流量分配比例为20%,则第一架构类型的目标节点总分配20G流量,第二种架构类型的目标节点总分配20G流量,第三种架构类型的目标节点总分配10G流量,在架构类型下有多个目标节点的情况下,目标节点之间可以按一定规则分配流量,例如第一种架构类型下有5个目标节点,在每一个节点可以分配4G流量,或者按照每一个目标节点的剩余资源来分配流量,具体根据实际情况进行分配即可,在此不做过多限定。

[0099] 步骤S2055,当待分配流量仍有余量时,从流量分配等级中选取下一个流量分配等级,并利用下一个流量分配等级对应的目标节点对余量进行分配,直至所有的流量分配完毕为止。

[0100] 具体的,当目标级别的所有目标节点流量均分配完毕,例如达到设定阈值,但待分配流量仍有余量,则从流量分配等级中选取下一个流量分配等级,例如下一个优先级的流量分配等级,确定下一个流量分配等级中每一个架构类型的目标节点的数量,利用公式1确定第一个流量分配等级中每一种架构类型的流量分配比例,将余量进行分配,直至所有的待分配流量均被分配完毕为止。

[0101] 通过上述方式,可以将目标节点根据流量分配等级划分为多个目标级别,分别统计每个目标节点内的每一个架构类型的数量和每一个架构类型下目标节点的数量,确定每一个流量分配等级的流量分配比例,当目标级别未能够把所有的待分配流量分配完成时,再选下一个流量分配等级进行分配,可以灵活有序的对完成对流量的分配,平衡各个节点的压力,保证系统的稳定运行。

[0102] 在一种可选的实施方式中,从所有的流量分配等级中选取目标级别,包括:

[0103] 选取流量分配等级中优先级最高的流量分配等级作为目标级别。

[0104] 具体的,可以按照每一个流量分配等级中所有目标节点的剩余资源确定流量分配等级的优先级,选取流量分配等级中优先级最高的流量分配等级作为目标级别。

[0105] 在一种可选的实施方式中,在每一个目标节点创建目标数量的容器之后,还包括如图3所示的方法步骤:

[0106] 步骤S301,监测目标节点中每一个容器的状态。

[0107] 具体的,可以通过命令查看容器的状态,对容器进行健康检查,获取每一个容器的健康状态。

[0108] 步骤S302,当确定第一容器的状态为异常时,将第一容器从目标节点进行驱逐,并根据每一个目标节点的剩余容器数量、每一个目标节点的流量分配等级以及每一个目标节点的架构类型,重新对目标节点的容器进行流量分配。

[0109] 具体的,第一容器为目标节点中任一容器,当确定第一容器的状态为异常时,将第一容器从目标节点驱逐到其他节点,然后根据每一个目标节点的剩余容器数量、每一个目标节点的流量分配等级以及每一个目标节点的架构类型,重新对目标节点的容器进行流量分配,其他节点应该是不参与流量分配的目标节点。

[0110] 在一种可选的实施方式中,对容器进行健康检查可以在容器创建完成时进行检查,也可以自定义运行时间,在自定义的时间进行健康检查,健康检查可以包括容器的健康状态,也可以包括容器的流量分配是否已达到阈值,若达到流量阈值,也可以重新计算流量分配策略,保证所有的流量都可以及时进行分配。

[0111] 通过上述方式,可以实时监测容器的健康状态,对状态异常的容器及时进行处理,保证程序运行环境的稳定。

[0112] 为了使本发明提供的方法更加清楚,本发明还提供一种在具体场景下的实施例,例如在如图4所示的“一云多芯”应用场景下,指用一套云操作系统来管理不同架构的硬件服务器集群,包括多个架构类型的目标节点,例如超微半导体(Advanced Micro Devices,简称AMD)架构、进阶精简指令集机器(Advanced RISC Machine,简称ARM)架构、龙芯LoongArch架构等等,每一个目标节点下挂载多个容器(pod),用于根据需要发起“一云多芯”应用的创建容器请求,“一云多芯”应用可以设置是否开启智能流量切分策略,“一云多芯”应用控制器接收到用户的创建请求后,当确定开启智能流量切分策略时,获取每一个节点的CPU使用率,根据CPU使用率筛选出目标节点,并根据CPU使用里设置不同的流量分配等级,针对每一个流量分配等级,根据请求中的容器的目标数量对工作负载下的容器按照目标数量进行分组,对目标节点下大于目标数量的容器进行驱逐,在容器数小于目标数量的目标节点下创建容器,使所有目标节点下的容器数量均等于目标数量,检测程序(例如webhook)检测到pod的创建,根据pod的注解(是否开启流量切分策略)以及图1所示的流量分配方法过滤不符合要求节点,选择最优绑定节点与pod进行绑定并反馈pod的最终状态,直至所有的流量分配完成。

[0113] 具体的一个流量分配等级下的智能流量切分策略可以通过如下公式实现:

$$[0114] \quad a_1 = N_{\text{pod_AMD}} * N_{\text{node_AMD}} * 50\% \quad (\text{公式2})$$

$$[0115] \quad a_2 = N_{\text{pod_ARM}} * N_{\text{node_ARM}} * 30\% \quad (\text{公式3})$$

$$[0116] \quad a_3 = N_{\text{pod_LoongArch}} * N_{\text{node_LoongArch}} * 10\% \quad (\text{公式4})$$

$$[0117] \quad a_4 = N_{\text{pod_other}} * N_{\text{node_other}} * 10\% \quad (\text{公式5})$$

[0118] 其中, a_1 为AMD架构类型的流量分配比例, $N_{\text{pod_AMD}}$ 为AMD架构类型的每一个节点的pod数量, $N_{\text{node_AMD}}$ 为ADM架构类型的目标节点的数量,50%为AMD架构类型的预设权重; a_2 为ARM架构类型的流量分配比例, $N_{\text{pod_ARM}}$ 为ARM架构类型的每一个节点的pod数量, $N_{\text{node_ARM}}$ 为ARM架构类型的目标节点的数量,30%为ARM架构类型的预设权重; a_3 为LoongArch架构类型的流量分配比例, $N_{\text{pod_LoongArch}}$ 为LoongArch架构类型的每一个节点的pod数量, $N_{\text{node_LoongArch}}$ 为LoongArch架构类型的目标节点的数量,10%为LoongArch架构类型的预设权重; a_4 为其他架构类型的流量分配比例, $N_{\text{pod_other}}$ 为其他架构类型的每一个节点的pod数量, $N_{\text{node_other}}$ 为其他架构类型的目标节点的数量,10%为其他架构类型的预设权重, $N_{\text{pod_AMD}}$ 、 $N_{\text{pod_ARM}}$ 、 $N_{\text{pod_LoongArch}}$ 以及 $N_{\text{pod_other}}$ 相同,均为创建请求中pod的目标数量。

[0119] 以10G待分配流量为例,分配到每一种架构类型的流量可以用如下公式表示:

$$[0120] \quad Q_{\text{AMD}} = \frac{10G * a_1}{a_1 + a_2 + a_3 + a_4} \quad (\text{公式6})$$

$$[0121] \quad Q_{\text{ARM}} = \frac{10G * a_2}{a_1 + a_2 + a_3 + a_4} \quad (\text{公式7})$$

$$[0122] \quad Q_{\text{LoongArch}} = \frac{10G * a3}{a1 + a2 + a3 + a4} \quad (\text{公式 } 8)$$

$$[0123] \quad Q_{\text{other}} = \frac{10G * a4}{a1 + a2 + a3 + a4} \quad (\text{公式 } 9)$$

[0124] 其中, Q_{AMD} 为AMD架构类型的节点分配的流量, Q_{ARM} 为ARM架构类型的节点分配的流量, $Q_{\text{LoongArch}}$ 为LoongArch架构类型的节点分配流量, Q_{other} 为其他架构类型的节点分配的流量。

[0125] 一云多芯应用创建后, 网格策略控制器 (Meshpolicy controller) 监听到服务的创建事件, 根据是否开启流量智能切分策略, 为对应的服务创建出网格策略, 给service (服务) 创建出虚拟服务 (virtualservice) 和目标规则 (destinationrule) 资源, 同时为service打上标识其配置一云多芯应用的流量智能切分策略 (即如图1所示的流量分配方法)。在“一云多芯”应用创建后, 可以设置流量监测模块, 采用基于普罗米修斯 (Prometheus) 的监控流量统计和“一云多芯”应用自身的健康检查机制, 开启进行健康检查的就绪探针和存活探针, 来实时监测应用的pod状态。

[0126] 例如可以使用就绪探针 (ReadinessProbe), 进行容器就绪检查, 判断服务是否已经正常工作 (即查看容器是否准备好接受Http请求), 如果服务没有加载完成或工作异常, 服务所在的Pod的IP地址会从服务的节点中移除, 不会再接受或响应任何请求; 使用存活探针 (LivenessProbe) 在线检查机制, 判断容器是否处于正在运行状态, 当服务崩溃或者死锁等情况发生时, 容器编排引擎 (kubernetes, 简称k8s) 会删除容器, 然后根据其设置的重启策略 (restart policy) 进行相关的重启操作, 若健康检查为正常状态 (例如健康检查请求 (例如HTTPGetAction类请求) 通过容器的IP、端口、PATH路径, 用HTTP GET方式请求进行检查, 返回响应码为 $200 \leq \text{响应码} < 400$), 若健康检查基于restart policy重启后仍异常, 则健康检查结果为异常, pod状态置为异常。当pod状态正常且监控流量达到pod流量阈值时, 可以触发流量智能切分机制, 使用图1所示的流量分配方法进行流量分配, 或者, 当健康检查机制将pod置为异常后, 也使用图1所示的流量分配方法进行流量分配。

[0127] 在本实施例中还提供了一种流量分配装置, 该装置用于实现上述实施例及优选实施方式, 已经进行过说明的不再赘述。如以下所使用的, 术语“模块”可以实现预定功能的软件和/或硬件的组合。尽管以下实施例所描述的装置较佳地以软件来实现, 但是硬件, 或者软件和硬件的组合的实现也是可能并被构想的。

[0128] 本实施例提供一种流量分配装置, 如图5所示, 包括:

[0129] 获取模块501, 用于获取容器创建请求, 其中, 容器创建请求中包括待创建容器的目标数量;

[0130] 查询模块502, 用于根据容器创建请求查询容器云中所有节点的剩余资源以及每一个节点的架构类型;

[0131] 筛选模块503, 用于根据每一个节点的剩余资源从容器云中所有的节点中筛选出目标节点;

[0132] 确定模块504, 用于根据每一个目标节点的剩余资源以及剩余资源与流量分配等级的映射关系, 确定每一个目标节点的流量分配等级;

[0133] 第一分配模块505,用于当确定目标节点包含的容器数量满足目标数量时,根据目标数量、每一个目标节点的流量分配等级以及每一个目标节点的架构类型,对目标节点的容器进行流量分配。

[0134] 在一些可选的实施方式中,筛选模块503中的剩余资源包括中央处理器的使用率,筛选模块503包括:

[0135] 确定单元,用于当第一节点的中央处理器的使用率小于或者等于预设阈值时,确定第一节点为目标节点,其中,第一节点为容器云中任一个节点。

[0136] 在一些可选的实施方式中,当确定目标节点包含的容器数量不满足目标数量时,装置还包括:

[0137] 驱逐模块506,用于当容器数量大于目标数量时,将大于目标数量的容器进行驱逐,使目标节点的容器数量等于目标数量;

[0138] 或者,

[0139] 新建模块507,用于当容器数量小于目标数量时,在目标节点新建容器,使目标节点的容器数量等于目标数量。

[0140] 在一些可选的实施方式中,流量分配等级包括至少一个,分配模块505,包括:

[0141] 选取单元,用于从所有的流量分配等级中选取目标级别;

[0142] 确定单元,用于确定目标级别中属于第一架构类型的目标节点数量,其中,第一架构类型为目标级别对应的目标节点所属的架构类型中任一个架构类型;根据目标数量,第一架构类型的目标节点数量,确定第一流量分配等级中第一架构类型的流量分配比例;

[0143] 流量分配单元,用于依次按照目标级别中每一种架构类型分别对应的流量分配比例,将待分配流量分配至目标级别分别对应的目标节点上;当待分配流量仍有余量时,从流量分配等级中选取下一个流量分配等级,并利用下一个流量分配等级对应的目标节点对余量进行分配,直至所有的流量分配完毕为止。

[0144] 在一些可选的实施方式中,选取单元,具体包括:

[0145] 选取流量分配等级中优先级最高的流量分配等级作为目标级别。

[0146] 在一些可选的实施方式中,确定单元中,根据目标数量,第一架构类型的目标节点数量,确定第一流量分配等级中第一架构类型的流量分配比例,具体通过如下公式确定:

$$[0147] \quad L_1 = N_{pod} * N_{node1} * m1$$

[0148] 其中, L_1 为第一架构类型的流量分配比例, N_{pod} 为目标数量, N_{node1} 为第一架构类型的目标节点数量, $m1$ 为第一架构类型的预设权重。

[0149] 在一些可选的实施方式中,在每一个目标节点创建目标数量的容器之后,装置还包括:

[0150] 监测模块,用于监测目标节点中每一个容器的状态;

[0151] 第二分配模块508,用于当第一容器的状态为异常时,将第一容器从目标节点进行驱逐,并根据每一个目标节点的剩余容器数量、每一个目标节点的流量分配等级以及每一个目标节点的架构类型,重新对目标节点的容器进行流量分配,其中,第一容器为目标节点中任一个容器。

[0152] 上述各个模块和单元的更进一步的功能描述与上述对应实施例相同,在此不再赘述。

[0153] 本实施例中的流量分配装置是以功能单元的形式来呈现,这里的单元是指ASIC (Application Specific Integrated Circuit,专用集成电路)电路,执行一个或多个软件或固定程序的处理器和存储器,和/或其他可以提供上述功能的器件。

[0154] 本发明实施例还提供一种计算机设备,具有上述图5所示的流量分配装置。

[0155] 请参阅图6,图6是本发明可选实施例提供的一种计算机设备的结构示意图,如图6所示,该计算机设备包括:一个或多个处理器10、存储器20,以及用于连接各部件的接口,包括高速接口和低速接口。各个部件利用不同的总线互相通信连接,并且可以被安装在公共主板上或者根据需要以其它方式安装。处理器可以对在计算机设备内执行的指令进行处理,包括存储在存储器中或者存储器上以在外部输入/输出装置(诸如,耦合至接口的显示设备)上显示GUI的图形信息的指令。在一些可选的实施方式中,若需要,可以将多个处理器和/或多条总线与多个存储器和多个存储器一起使用。同样,可以连接多个计算机设备,各个设备提供部分必要的操作(例如,作为服务器阵列、一组刀片式服务器、或者多处理器系统)。图6中以一个处理器10为例。

[0156] 处理器10可以是中央处理器,网络处理器或其组合。其中,处理器10还可以进一步包括硬件芯片。上述硬件芯片可以是专用集成电路,可编程逻辑器件或其组合。上述可编程逻辑器件可以是复杂可编程逻辑器件,现场可编程逻辑门阵列,通用阵列逻辑或其任意组合。

[0157] 其中,存储器20存储有可由至少一个处理器10执行的指令,以使至少一个处理器10执行实现上述实施例示出的方法。

[0158] 存储器20可以包括存储程序区和存储数据区,其中,存储程序区可存储操作系统、至少一个功能所需要的应用程序;存储数据区可存储根据计算机设备的使用所创建的数据等。此外,存储器20可以包括高速随机存取存储器,还可以包括非瞬时存储器,例如至少一个磁盘存储器件、闪存器件、或其他非瞬时固态存储器件。在一些可选的实施方式中,存储器20可选包括相对于处理器10远程设置的存储器,这些远程存储器可以通过网络连接至该计算机设备。上述网络的实例包括但不限于互联网、企业内部网、局域网、移动通信网及其组合。

[0159] 存储器20可以包括易失性存储器,例如,随机存取存储器;存储器也可以包括非易失性存储器,例如,快闪存储器,硬盘或固态硬盘;存储器20还可以包括上述种类的存储器的组合。

[0160] 该计算机设备还包括通信接口30,用于该计算机设备与其他设备或通信网络通信。

[0161] 本发明实施例还提供了一种计算机可读存储介质,上述根据本发明实施例的方法可在硬件、固件中实现,或者被实现为可记录在存储介质,或者被实现通过网络下载的原始存储在远程存储介质或非暂时机器可读存储介质中并将被存储在本地存储介质中的计算机代码,从而在此描述的方法可被存储在使用通用计算机、专用处理器或者可编程或专用硬件的存储介质上的这样的软件处理。其中,存储介质可为磁碟、光盘、只读存储记忆体、随机存储记忆体、快闪存储器、硬盘或固态硬盘等;进一步地,存储介质还可以包括上述种类的存储器的组合。可以理解,计算机、处理器、微处理器控制器或可编程硬件包括可存储或接收软件或计算机代码的存储组件,当软件或计算机代码被计算机、处理器或硬件访问且

执行时,实现上述实施例示出的方法。

[0162] 虽然结合附图描述了本发明的实施例,但是本领域技术人员可以在不脱离本发明的精神和范围的情况下做出各种修改和变型,这样的修改和变型均落入由所附权利要求所限定的范围之内。

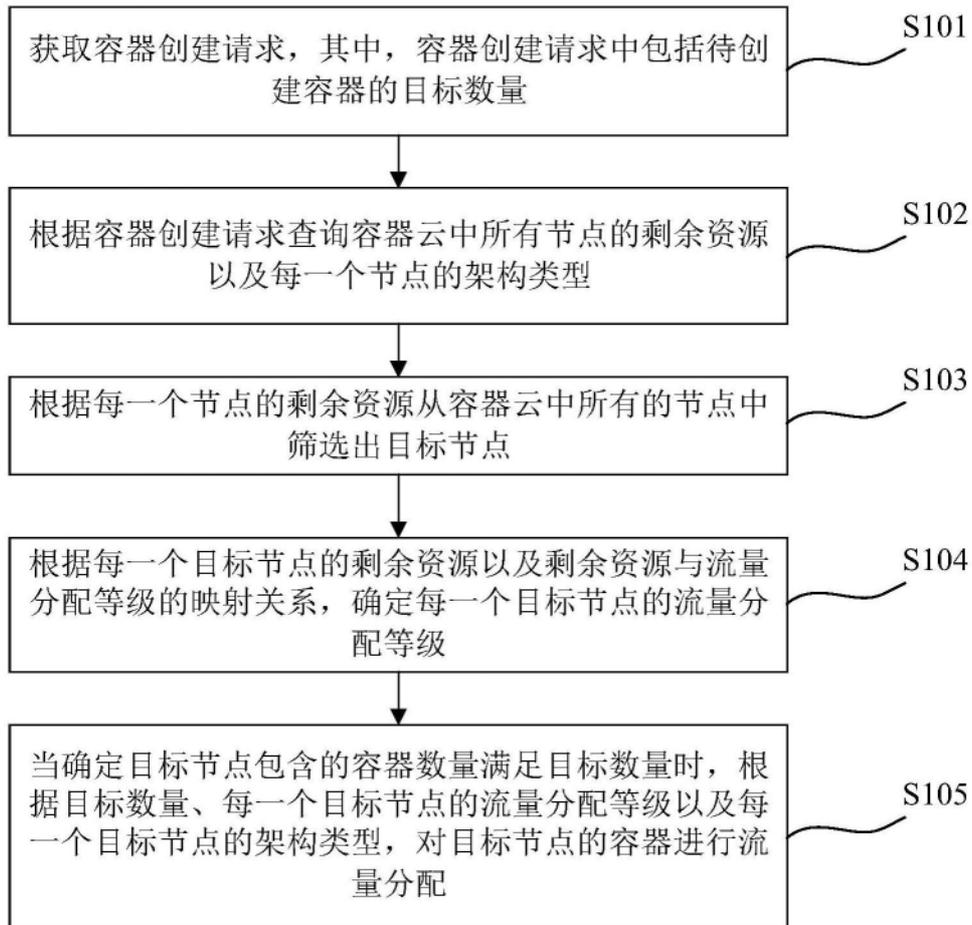


图1

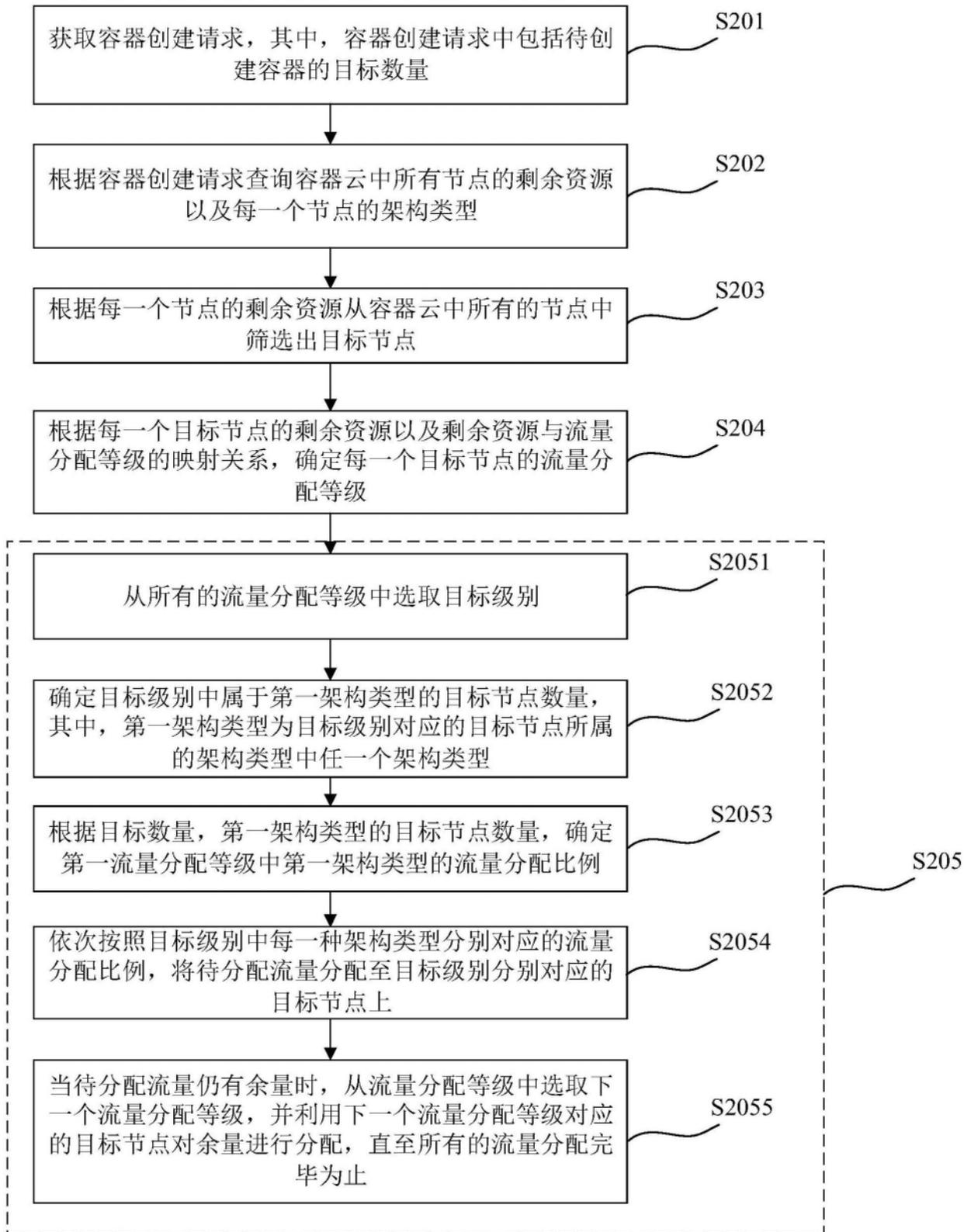


图2

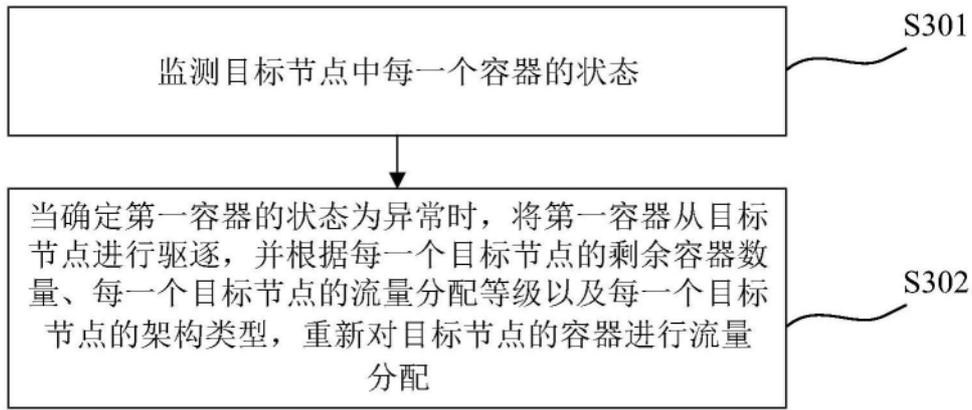


图3

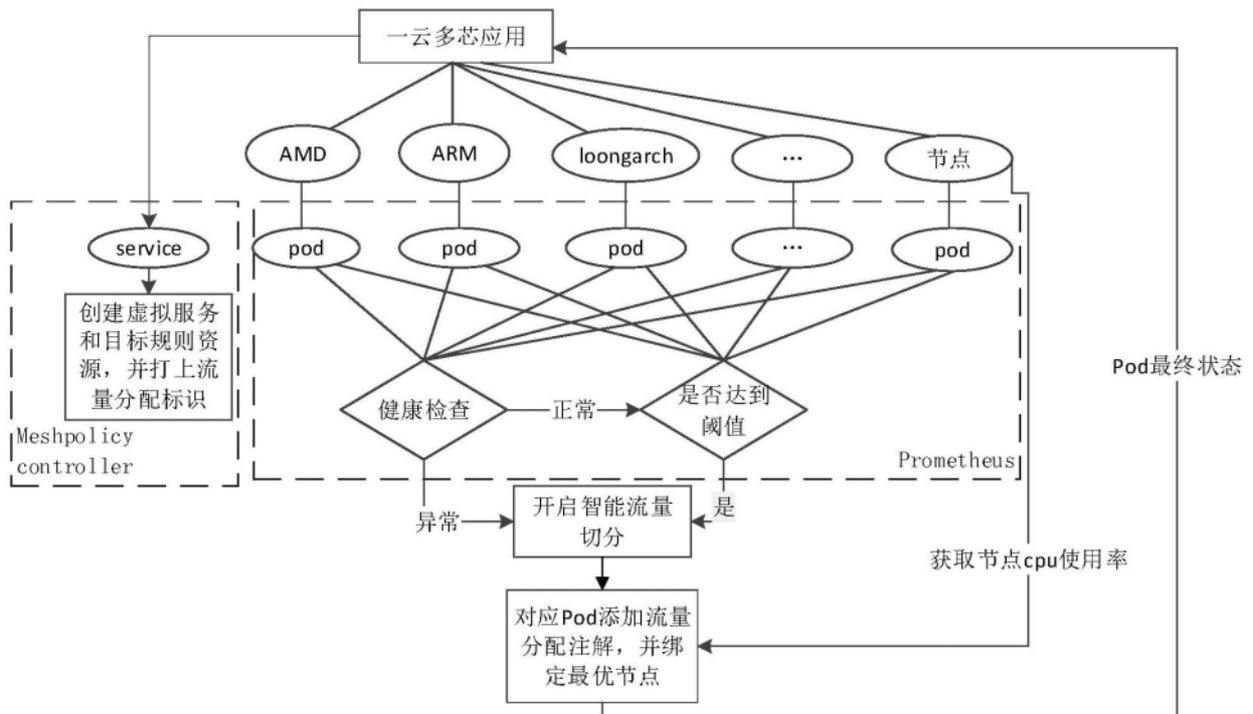


图4

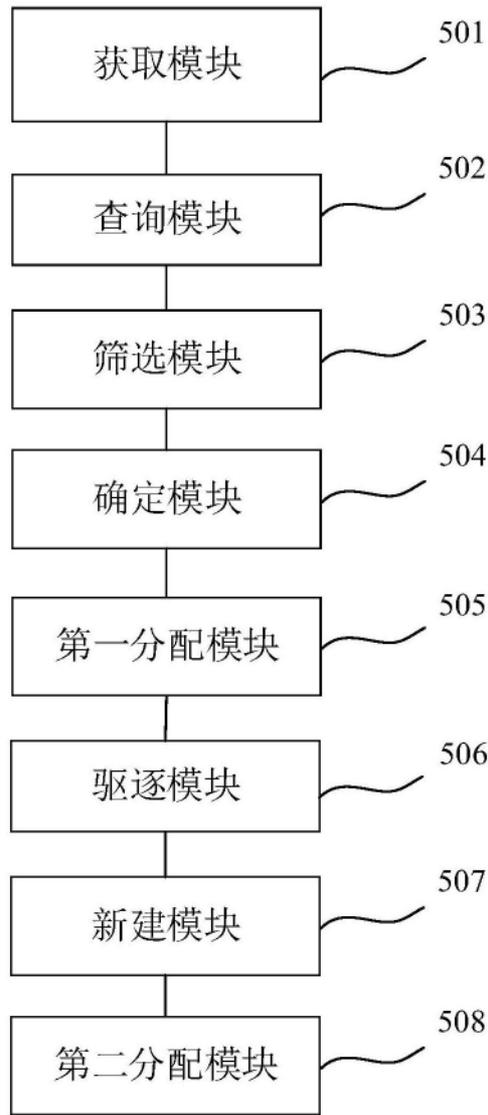


图5

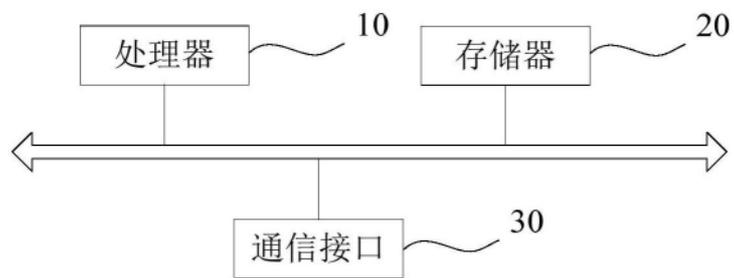


图6