



(12) 发明专利

(10) 授权公告号 CN 114710549 B

(45) 授权公告日 2023. 09. 22

(21) 申请号 202210179642.7

G06F 9/455 (2006.01)

(22) 申请日 2022.02.25

H04L 12/02 (2006.01)

(65) 同一申请的已公布的文献号

H04L 45/74 (2022.01)

申请公布号 CN 114710549 A

H04L 61/5007 (2022.01)

(43) 申请公布日 2022.07.05

(56) 对比文件

(73) 专利权人 网宿科技股份有限公司

CN 113268308 A, 2021.08.17

地址 200030 上海市徐汇区斜土路2899号

CN 113709810 A, 2021.11.26

光启文化广场A幢5楼

CN 113760452 A, 2021.12.07

(72) 发明人 李竞佳 李强达

CN 113949707 A, 2022.01.18

US 2022012045 A1, 2022.01.13

(74) 专利代理机构 北京华智则铭知识产权代理有限公司 11573

冯文超. 容器云平台网络资源配置管理系统的设计. 工业仪表与自动化装置. 2018, (第01期), 全文.

专利代理师 王昌贵

审查员 林群芳

(51) Int. Cl.

H04L 67/306 (2022.01)

H04L 67/63 (2022.01)

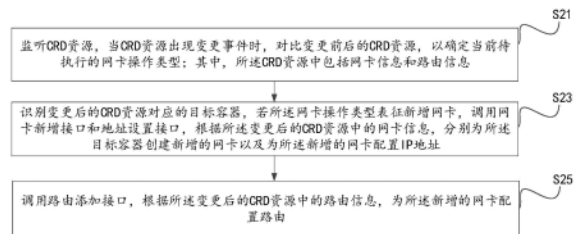
权利要求书2页 说明书9页 附图4页

(54) 发明名称

一种容器平台中网卡的动态管理方法、系统及业务节点

(57) 摘要

本发明公开了一种容器平台中网卡的动态管理方法、系统及业务节点,其中,所述方法包括:监听CRD资源,当CRD资源出现变更事件时,对比变更前后的CRD资源,以确定当前待执行的网卡操作类型;其中,所述CRD资源中包括网卡信息和路由信息;识别变更后的CRD资源对应的目标容器,若所述网卡操作类型表征新增网卡,调用网卡新增接口和地址设置接口,根据所述变更后的CRD资源中的网卡信息,分别为所述目标容器创建新增的网卡以及为所述新增的网卡配置IP地址;调用路由添加接口,根据所述变更后的CRD资源中的路由信息,为所述新增的网卡配置路由。本申请提供的技术方案,能够在POD运行过程中对网卡进行动态管理。



1. 一种容器平台中网卡的动态管理方法,其特征在于,所述方法应用于容器平台的业务节点中,所述方法包括:

在管理容器启动时,调用第一容器网络接口,以通过所述第一容器网络接口配置内网网卡;

链式调用第二容器网络接口,以通过所述第二容器网络接口将POD信息传递给地址管理客户端;

通过所述地址管理客户端,根据所述POD信息识别容器创建请求对应的CRD资源,并从所述CRD资源中获取配置信息,以根据所述配置信息为POD创建网卡;

监听CRD资源,当CRD资源出现变更事件时,对比变更前后的CRD资源,以确定当前待执行的网卡操作类型;其中,所述CRD资源中包括网卡信息和路由信息;

识别变更后的CRD资源对应的目标容器,若所述网卡操作类型表征新增网卡,调用网卡新增接口和地址设置接口,根据所述变更后的CRD资源中的网卡信息,分别为所述目标容器创建新增的网卡以及为所述新增的网卡配置IP地址;

调用路由添加接口,根据所述变更后的CRD资源中的路由信息,为所述新增的网卡配置路由。

2. 根据权利要求1所述的方法,其特征在于,识别变更后的CRD资源对应的目标容器包括:

从变更后的CRD资源中获取空间标识和选择标签,并调用POD的客户端接口,在所述空间标识指向的用户空间中查询与所述选择标签相匹配的POD对象;

获取所述POD对象的容器标识,并将所述容器标识指向的容器作为识别出的目标容器。

3. 根据权利要求1或2所述的方法,其特征在于,在识别变更后的CRD资源对应的目标容器之后,所述方法还包括:

查询所述目标容器的进程号,并根据所述进程号进入所述目标容器的内部空间;

在所述目标容器的内部空间内操作静态路由文件,以通过所述静态路由文件将变更后的CRD资源中的配置信息持久化到文件系统中。

4. 根据权利要求1所述的方法,其特征在于,所述方法还包括:

若所述网卡操作类型表征删除网卡,从所述变更后的CRD资源的网卡信息中识别出待删除的目标网卡,调用网卡删除接口和地址删除接口,删除所述目标网卡以及删除所述目标网卡的IP地址;

从所述变更后的CRD资源的路由信息中识别出待删除的目标路由,调用路由删除接口,删除所述目标路由。

5. 根据权利要求1所述的方法,其特征在于,所述方法还包括:

若所述网卡操作类型表征修改网卡,从所述变更后的CRD资源的网卡信息中识别出修改前的第一网卡和修改后的第二网卡;

在删除所述第一网卡,并且删除所述第一网卡对应的IP地址和路由之后,添加所述第二网卡,并添加所述第二网卡对应的IP地址和路由。

6. 根据权利要求1所述的方法,其特征在于,在管理容器启动之前,所述方法还包括:

在主节点接收到容器创建请求,并将所述容器创建请求对应的CRD资源存储到数据库后,接收所述主节点中的调度器发来的POD网络创建指令。

7. 根据权利要求1所述的方法,其特征在于,所述POD信息中至少包括POD标识;根据所述POD信息识别所述容器创建请求对应的CRD资源包括:

根据所述POD标识查询对应的所属资源字段,并从所述所属资源字段中识别CRD标识,所述CRD标识指向的CRD资源作为所述容器创建请求对应的CRD资源。

8. 根据权利要求1所述的方法,其特征在于,在根据所述配置信息为POD创建网卡之后,所述方法还包括:

将创建网卡的结果反馈给所述第二容器网络接口,若所述创建网卡的结果表征创建成功,继续创建POD中的业务容器。

9. 一种容器平台中网卡的动态管理系统,其特征在于,所述系统包括:

资源识别单元,用于监听CRD资源,当CRD资源出现变更事件时,对比变更前后的CRD资源,以确定当前待执行的网卡操作类型;其中,所述CRD资源中包括网卡信息和路由信息;其中,在监听CRD资源之前,包括:在管理容器启动时,调用第一容器网络接口,以通过所述第一容器网络接口配置内网网卡;链式调用第二容器网络接口,以通过所述第二容器网络接口将POD信息传递给地址管理客户端;通过所述地址管理客户端,根据所述POD信息识别容器创建请求对应的CRD资源,并从所述CRD资源中获取配置信息,以根据所述配置信息为POD创建网卡;

网络设置单元,用于识别变更后的CRD资源对应的目标容器,若所述网卡操作类型表征新增网卡,调用网卡新增接口和地址设置接口,根据所述变更后的CRD资源中的网卡信息,分别为所述目标容器创建新增的网卡以及为所述新增的网卡配置IP地址;

路由设置单元,用于调用路由添加接口,根据所述变更后的CRD资源中的路由信息,为所述新增的网卡配置路由。

10. 一种容器平台的业务节点,其特征在于,所述业务节点包括存储器和处理器,所述存储器用于存储计算机程序,所述计算机程序被所述处理器执行时,实现如权利要求1至8中任一所述的方法。

一种容器平台中网卡的动态管理方法、系统及业务节点

技术领域

[0001] 本发明涉及互联网技术领域,特别涉及一种容器平台中网卡的动态管理方法、系统及业务节点。

背景技术

[0002] Kubernetes提供了应用部署、调度、更新、维护的一种机制,但它在Pod-to-Pod的通信网络上还缺少一个普适的解决方案。在容器部署中,CNI(Container Network Interface,容器网络接口)为容器集群工具(Kubernetes、Mesos、OpenShift等)提供了一个网络标准。基于此CNI标准,当前涌现了一大批CNI插件(Cilium、Calico、Flannel等解决方案),但这些CNI插件主要方向都是为了解决网络的互通或性能,而在多网卡的管理上显得乏力。

[0003] 传统上,网络功能使用多个网络接口将数据网络平面与管理网络平面分离控制,从而支持不同的协议,满足不同的调整和配置要求。当前,Kubernetes缺乏支持VNF(Virtual Network Function,虚拟化网络功能)中多个网络接口的所需功能。基于此,目前Intel提供了Multus方案,其中提供了将多个接口添加到POD容器组)的功能。

[0004] Multus方案虽然能创建多网卡,但其在动态管理仍显不足,网卡创建的时机是在POD启动时。而当POD运行时,无法对网卡进行动态的管理,而且也无法对网卡上的各项信息进行动态配置。

发明内容

[0005] 本申请的目的在于提供一种容器平台中网卡的动态管理方法、系统及业务节点,能够在POD运行过程中对网卡进行动态管理。

[0006] 为实现上述目的,本申请一方面提供一种容器平台中网卡的动态管理方法,所述方法应用于容器平台的业务节点中,所述方法包括:监听CRD资源,当CRD资源出现变更事件时,对比变更前后的CRD资源,以确定当前待执行的网卡操作类型;其中,所述CRD资源中包括网卡信息和路由信息;识别变更后的CRD资源对应的目标容器,若所述网卡操作类型表征新增网卡,调用网卡新增接口和地址设置接口,根据所述变更后的CRD资源中的网卡信息,分别为所述目标容器创建新增的网卡以及为所述新增的网卡配置IP地址;调用路由添加接口,根据所述变更后的CRD资源中的路由信息,为所述新增的网卡配置路由。

[0007] 在一个实施方式中,识别变更后的CRD资源对应的目标容器包括:从变更后的CRD资源中获取空间标识和选择标签,并调用POD的客户端接口,在所述空间标识指向的用户空间中查询与所述选择标签相匹配的POD对象;获取所述POD对象的容器标识,并将所述容器标识指向的容器作为识别出的目标容器。

[0008] 在一个实施方式中,在识别变更后的CRD资源对应的目标容器之后,所述方法还包括:查询所述目标容器的进程号,并根据所述进程号进入所述目标容器的内部空间;在所述目标容器的内部空间内操作静态路由文件,以通过所述静态路由文件将变更后的CRD资源

中的配置信息持久化到文件系统中。

[0009] 在一个实施方式中,所述方法还包括:若所述网卡操作类型表征删除网卡,从所述变更后的CRD资源的网卡信息中识别出待删除的目标网卡,调用网卡删除接口和地址删除接口,删除所述目标网卡以及删除所述目标网卡的IP地址;从所述变更后的CRD资源的路由信息中识别出待删除的目标路由,调用路由删除接口,删除所述目标路由。

[0010] 在一个实施方式中,所述方法还包括:若所述网卡操作类型表征修改网卡,从所述变更后的CRD资源的网卡信息中识别出修改前的第一网卡和修改后的第二网卡;在删除所述第一网卡,并且删除所述第一网卡对应的IP地址和路由之后,添加所述第二网卡,并添加所述第二网卡对应的IP地址和路由。

[0011] 在一个实施方式中,在监听CRD资源之前,所述方法还包括:在主节点接收到容器创建请求,并将所述容器创建请求对应的CRD资源存储到数据库后,接收所述主节点中的调度器发来的POD网络创建指令;在管理容器启动时,调用第一容器网络接口,以通过所述第一容器网络接口配置内网网卡;链式调用第二容器网络接口,以通过所述第二容器网络接口将POD信息传递给地址管理客户端;通过所述地址管理客户端,根据所述POD信息识别所述容器创建请求对应的CRD资源,并从所述CRD资源中获取配置信息,以根据所述配置信息为POD创建网卡。

[0012] 在一个实施方式中,所述POD信息中至少包括POD标识;根据所述POD信息识别所述容器创建请求对应的CRD资源包括:根据所述POD标识查询对应的所属资源字段,并从所述所属资源字段中识别CRD标识,所述CRD标识指向的CRD资源作为所述容器创建请求对应的CRD资源。

[0013] 在一个实施方式中,在根据所述配置信息为POD创建网卡之后,所述方法还包括:将创建网卡的结果反馈给所述第二容器网络接口,若所述创建网卡的结果表征创建成功,继续创建POD中的业务容器。

[0014] 为实现上述目的,本申请另一方面还提供一种容器平台中网卡的动态管理系统,所述系统包括:资源识别单元,用于监听CRD资源,当CRD资源出现变更事件时,对比变更前后的CRD资源,以确定当前待执行的网卡操作类型;其中,所述CRD资源中包括网卡信息和路由信息;网络设置单元,用于识别变更后的CRD资源对应的目标容器,若所述网卡操作类型表征新增网卡,调用网卡新增接口和地址设置接口,根据所述变更后的CRD资源中的网卡信息,分别为所述目标容器创建新增的网卡以及为所述新增的网卡配置IP地址;路由设置单元,用于调用路由添加接口,根据所述变更后的CRD资源中的路由信息,为所述新增的网卡配置路由。

[0015] 为实现上述目的,本申请另一方面还提供一种容器平台的业务节点,所述业务节点包括存储器和处理器,所述存储器用于存储计算机程序,所述计算机程序被所述处理器执行时,实现上述的容器平台中网卡的动态管理方法。

[0016] 本申请提供的技术方案,在容器创建之后,可以监听容器的CRD(Custom Resource Definition,用户资源自定义)资源,当该CRD资源发生变更事件时,就表明容器的网卡配置发生了更改。例如可以在容器下新增网卡、删除网卡,或者对已有的网卡进行修改。通过对比变更前后的CRD资源,可以确定出网卡操作类型。如果网卡操作类型表明在容器下新增网卡,可以根据变更后的CRD资源中的网卡信息和路由信息,调用对应的接口,为容器新增网

卡、配置IP地址以及配置路由,从而达到动态管理网卡的目的。

[0017] 进一步地,容器平台在创建容器网络时,会链式调用CNI插件,鉴于此,本申请调用的第一链可以是现有的CNI插件,第二链可以是自研的CNI插件,这样即能保留POD原有的网络能力,又能增强了POD的多网卡功能。

附图说明

[0018] 为了更清楚地说明本发明实施方式中的技术方案,下面将对实施方式描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图仅仅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。

[0019] 图1是本发明实施方式中kubernetes集群的结构示意图;

[0020] 图2是本发明实施方式中为POD配置网络环境的步骤示意图;

[0021] 图3是本发明实施方式中链式调用CNI插件的示意图;

[0022] 图4是本发明实施方式中动态管理网卡的步骤示意图;

[0023] 图5是本发明实施方式中动态管理系统的功能模块示意图;

[0024] 图6是本发明实施方式中业务节点的结构示意图;

[0025] 图7是本发明实施方式中计算机终端的结构示意图。

具体实施方式

[0026] 为使本申请的目的、技术方案和优点更加清楚,下面将结合本申请具体实施方式及相应的附图对本申请技术方案进行清楚、完整地描述。显然,所描述的实施方式仅是本申请一部分实施方式,而不是全部的实施方式。基于本申请中的实施方式,本领域普通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方式,都属于本申请保护的范围。

[0027] 本申请提供的容器平台中网卡的动态管理方法,可以应用于图1所示的kubernetes容器平台中。在kubernetes中,可以包括主节点(master)和业务节点(node)。其中,主节点可以接收业务方发起的容器创建请求,并响应于该请求,选择一个业务节点,由选择的业务节点进行POD创建和网络配置的过程。

[0028] 具体地,请参阅图1和图2,为POD配置网络的过程可以包括以下多个步骤。

[0029] S11:在主节点接收到容器创建请求,并将所述容器创建请求对应的CRD资源存储到数据库后,接收所述主节点中的调度器发来的POD网络创建指令。

[0030] 在本实施方式中,可以根据kubernetes的标准自定义资源的方法,创建一种新的CRD对象,该新的CRD对象中的配置信息(networkInfos)可以包括网卡信息(nic)和路由信息(route)。其中,网卡信息可以包括网卡名(name)、MAC地址(mac)、IP地址(address)以及网关(gateway)。路由信息可以包括目标地址(dst)和网关(gateway)。

[0031] 在一个具体应用示例中,CRD对象中包含的配置信息可以如下所示:

networkInfos:

- nic:
- name: eth1
- mac: “02:00:00:00:00:11”
- addresses:
- [0032] - address: “11.11.11.11/24”
- gateway: “11.11.11.1”
- route:
- dst: “33.33.33.0/24”
- gateway: “22.22.22.1”

[0033] 在本实施方式中,如果业务方要在容器平台中为POD配置网络,可以按照CRD对象中的数据格式,调用容器平台的API接口,向容器平台发起容器创建请求。在该容器创建请求中,可以携带按照CRD对象中的数据格式进行编写的CRD资源。在一个实际应用示例中,业务方可以通过kubect1 apply-f crd.yaml这样的命令,向容器平台发起容器创建请求。

[0034] 容器平台的主节点中的api-server接收到该容器创建请求后,可以将其中的CRD资源存储到数据库(etcd)中,同时,主节点内的CRD控制器(CRD Controller)监听到CRD资源后,可以告知主节点中的调度器(scheduler),调度器便可以选择一个业务节点,并通知该业务节点中的kubect1进程创建POD网络。

[0035] 在本实施方式中,业务节点中的kubect1进程接收到调度器发来的POD网络创建指令后,可以识别该POD网络创建指令中携带的POD信息。通常而言,该POD信息可以表示为<namespace,podname,containerId,netns>,其中,namespace表示POD所在的用户空间的空间标识,podname表示准备进行网络配置的POD的标识,containerId则表示POD中管理容器的容器标识,netns(network namespace)则表示待创建的POD网络对应的网络标识。

[0036] 需要说明的是的,在一个POD中,通常可以包括一个管理容器和多个业务容器,在业务容器启动之前,通过为管理容器配置网卡,可以搭建起POD的网络环境。这样,在业务容器启动之后,便可以识别出相应的网卡,进而利用搭建起的网络环境向外提供业务。本申请中描述的为容器配置网卡的过程,指的都是为管理容器配置网卡。

[0037] S13:在管理容器启动时,调用第一容器网络接口,以通过所述第一容器网络接口配置内网网卡。

[0038] S15:链式调用第二容器网络接口,以通过所述第二容器网络接口将POD信息传递给地址管理客户端。

[0039] 在本实施方式中,kubect1在创建POD网络时,可以链式调用CNI。具体地,请参阅图3,在管理容器(pause容器)启动时,kubect1进程可以调用第一CNI,该第一CNI可以负责配置内网网卡eth0,该内网网卡eth0是用于kubernetes管理POD时使用的。

[0040] 在第一CNI调用完成后,kubect1可以链式调用第二CNI,该第二CNI所实现的功能可以是本申请中自定义的。具体地,第二CNI被调用后,可以将POD网络创建指令中携带的

POD信息传递给地址管理客户端(IPAM Agent),也就是说,可以将<namespace,podname,containerId,netns>这样的POD信息传递给地址管理客户端。

[0041] S17:通过所述地址管理客户端,根据所述POD信息识别所述容器创建请求对应的CRD资源,并从所述CRD资源中获取配置信息,以根据所述配置信息为POD创建网卡。

[0042] 在本实施方式中,地址管理客户端接收到传递来的POD信息后,可以识别其中的POD标识。在kubernetes中,POD标识与所属资源字段(ownerReference)具备对应关系,根据识别出的POD标识,可以查询到对应的所属资源字段。在所属资源字段中,会携带POD对应的CRD资源的CRD标识。这样,根据CRD标识,可以读取到对应的CRD资源。

[0043] 在本实施方式中,在CRD资源中可以包含步骤S11中描述的配置信息(networkInfos),从该配置信息中,可以识别出配置信息。该配置信息例如可以包括网卡名称和MAC地址等。这样,基于CRD资源中的配置信息,地址管理客户端便可以为POD中的管理容器创建一张或者多张网卡。

[0044] 在创建网卡之后,地址管理客户端可以将创建网卡的结果反馈给第二CNI,这样,kubelet进程便可以获知本次创建网卡的结果。如果创建网卡的结果表征创建成功,则表明POD内具备了对外提供业务的网络环境,此时kubelet可以继续创建POD中的业务容器。当业务容器启动后,就可以识别出相应的网卡,并对外提供业务。而如果创建网卡的结果表征创建失败,那么就可以按照默认逻辑进行处理,例如可以重新尝试创建网卡,或者重新尝试创建管理容器。

[0045] 在本申请中,之所以采用链式调用的方式,是因为目前大部分的CNI插件都只能做到管理网络与业务网络共用同一张网卡,而对于有些应用场景需要分离管理网络与数据网络分离,单一的CNI插件则很难灵活地到这一点。通过本申请链式调用第一CNI插件和第二CNI插件的方式,可以做到管理网络与业务网络的网卡分离,并且可以为业务网络配置多个网卡,在保证数据安全的情况下,极大地提高了网络部署的灵活度。

[0046] 在本申请中,在为POD配置了网络环境后,还可以在容器运行过程中,动态地对其中的网卡进行管理。具体地请参阅图4,本申请一个实施方式提供的容器平台中网卡的动态管理方法,可以包括以下多个步骤。

[0047] S21:监听CRD资源,当CRD资源出现变更事件时,对比变更前后的CRD资源,以确定当前待执行的网卡操作类型;其中,所述CRD资源中包括网卡信息和路由信息。

[0048] 在本实施方式中,每个业务节点上都可以通过地址管理客户端来监听CRD资源,如果POD的网络环境发生变化(新增网卡、修改网卡信息、删除网卡等),那么对应的CRD资源就会出现变更事件(update事件)。地址管理客户端在监听到该事件后,可以根据变更后的CRD资源,对POD中的网卡执行对应的操作。

[0049] 具体地,当用户修改CRD资源的编排时,CRD控制器会发出update事件,地址管理客户端通过调用CRD的客户端接口可以监听CRD资源的变更事件。例如,在一个具体应用示例中,用户在CRD资源中添加了一张名称为eth2的网卡,更改后的CRD资源可以如下所示:


```
networkInfos:
  - nic:
[0050]   - name: eth1
      mac: "02:00:00:00:00:11"
      addresses:
        address: "11.11.11.11/24"
        gateway: "11.11.11.1"
  - name: eth2
[0051]   mac: "02:00:00:00:00:22"
      addresses:
        address: "22.22.22.22/24"
        gateway: "22.22.22.1"
```

[0052] 其中,名称为eth1的网卡是在为POD配置网络环境时创建的,名称为eth2的网卡是后续用户添加的。

[0053] 在本实施方式中,地址管理客户端监听到CRD资源的变更事件后,可以对比变更前后的CRD资源,从而可以明确当前待执行的网卡操作类型。例如,在上述的应用示例中,待执行的网卡操作类型就是新增网卡。在实际应用中,待执行的网卡操作类型还可以是删除网卡、修改网卡等。

[0054] S23:识别变更后的CRD资源对应的目标容器,若所述网卡操作类型表征新增网卡,调用网卡新增接口和地址设置接口,根据所述变更后的CRD资源中的网卡信息,分别为所述目标容器创建新增的网卡以及为所述新增的网卡配置IP地址。

[0055] S25:调用路由添加接口,根据所述变更后的CRD资源中的路由信息,为所述新增的网卡配置路由。

[0056] 在本实施方式中,地址管理客户端根据变更后的CRD资源,可以识别出应当对哪个容器进行网卡操作,该需要进行网卡操作的容器便可以是CRD资源对应的目标容器。

[0057] 具体地,变更后的CRD资源中可以包括空间标识(namespace)和选择标签(selectorLabels),其中,空间标识可以限定POD所在的用户空间,而选择标签则可以在用户空间中限定具体的POD对象。在本实施方式中,从变更后的CRD资源中获取空间标识和选择标签后,可以调用POD的客户端接口,在空间标识指向的用户空间中查询与选择标签相匹配的POD对象。例如,可以通过以下指令来调用POD的客户端接口:

```
[0058] podsLister.Pods(namespace).List(selectorLabels)
```

[0059] 在查询到相匹配的POD对象后,可以获取所述POD对象的容器标识(containerId),该容器标识可以指向POD对象中的管理容器,这样,所述容器标识指向的容器便可以作为需要执行网卡操作的目标容器。

[0060] 在一个实施方式向中,可以将变更后的CRD资源中的配置信息持久化到管理容器的内容,这样,在管理容器重新启动之后,依然可以按照变更后的CRD资源中的配置信息进行

行网卡配置。

[0061] 具体地,在识别出目标容器之后,可以通过与containerd进程或者dockerd进程进行交互,从而查找到目标容器对应的进程号(pid),然后可以采用nsenter的方式,根据该进程号进入到目标容器的内部空间。在目标容器的内部空间内操作静态路由文件,可以通过静态路由文件将变更后的CRD资源中的配置信息持久化到文件系统中。

[0062] 在一个具体应用场景中,地址管理客户端可以操作目标容器内部的ifcfg-eth0\route-eth0\static-routes文件,将配置信息持久化到文件系统中去。这样,在目标容器重新启动之后,依然可以按照变更后的CRD资源中的配置信息进行网卡配置。

[0063] 在本实施方式中,如果待执行的网卡操作类型表征新增网卡,那么可以根据变更后的CRD资源中的网卡信息,调用网卡新增接口netlink.add,为目标容器创建新增的网卡。在网卡创建后,还可以调用地址设置接口netlink.addradd,为新增的网卡配置IP地址。

[0064] 在完成了网卡信息的配置后,还可以根据变更后的CRD资源中的路由信息,调用路由添加接口netlink.routeadd为新增的网卡配置路由。

[0065] 在一个实施方式中,如果网卡操作类型表征删除网卡,那么可以采用类型的方式,从变更后的CRD资源的网卡信息中识别出待删除的目标网卡,并调用网卡删除接口netlink.del和地址删除接口netlink.addrdel,删除目标网卡以及删除目标网卡的IP地址。然后,可以从变更后的CRD资源的路由信息中识别出待删除的目标路由,并调用路由删除接口netlink.routedel,删除目标路由。

[0066] 在一个实施方式中,如果网卡操作类型表征修改网卡,那么可以按照先将原始的网卡删除,再添加新增的网卡方式来实现。具体地,可以从变更后的CRD资源的网卡信息中识别出修改前的第一网卡和修改后的第二网卡,然后可以采用上述的删除网卡的操作过程,在删除第一网卡,并且删除第一网卡对应的IP地址和路由之后,再采用上述的新增网卡的操作过程,为目标容器添加第二网卡,并添加第二网卡对应的IP地址和路由,从而完成网卡修改的过程。

[0067] 本申请提供的技术方案,在容器创建之后,可以监听容器的CRD(Custom Resource Definition,用户资源自定义)资源,当该CRD资源发生变更事件时,就表明容器的网卡配置发生了更改。例如可以在容器下新增网卡、删除网卡,或者对已有的网卡进行修改。通过对比变更前后的CRD资源,可以确定出网卡操作类型。如果网卡操作类型表明在容器下新增网卡,可以根据变更后的CRD资源中的网卡信息和路由信息,调用对应的接口,为容器新增网卡、配置IP地址以及配置路由,从而达到动态管理网卡的目的。

[0068] 进一步地,容器平台在创建容器网络时,会链式调用CNI插件,鉴于此,本申请调用的第一链可以是现有的CNI插件,第二链可以是自研的CNI插件,这样即能保留POD原有的网络能力,又能增强了POD的多网卡功能。

[0069] 请参阅图5,本申请还提供一种容器平台中网卡的动态管理系统,所述系统包括:

[0070] 资源识别单元,用于监听CRD资源,当CRD资源出现变更事件时,对比变更前后的CRD资源,以确定当前待执行的网卡操作类型;其中,所述CRD资源中包括网卡信息和路由信息;

[0071] 网络设置单元,用于识别变更后的CRD资源对应的目标容器,若所述网卡操作类型表征新增网卡,调用网卡新增接口和地址设置接口,根据所述变更后的CRD资源中的网卡信

息,分别为所述目标容器创建新增的网卡以及为所述新增的网卡配置IP地址;

[0072] 路由设置单元,用于调用路由添加接口,根据所述变更后的CRD资源中的路由信息,为所述新增的网卡配置路由。

[0073] 请参阅图6,本申请还提供一种容器平台的业务节点,所述业务节点包括存储器和处理器,所述存储器用于存储计算机程序,所述计算机程序被所述处理器执行时,实现上述的容器平台中网卡的动态管理方法。

[0074] 请参阅图7,在本申请中,上述实施例中的技术方案可以应用于如图7所示的计算机终端10上。计算机终端10可以包括一个或多个(图中仅示出一个)处理器102(处理器102可以包括但不限于微处理器MCU或可编程逻辑器件FPGA等的处理装置)、用于存储数据的存储器104、以及用于通信功能的传输模块106。本领域普通技术人员可以理解,图7所示的结构仅为示意,其并不对上述电子装置的结构造成限定。例如,计算机终端10还可包括比图7中所示更多或者更少的组件,或者具有与图7所示不同的配置。

[0075] 存储器104可用于存储应用软件的程序以及模块,处理器102通过运行存储在存储器104内的软件程序以及模块,从而执行各种功能应用以及数据处理。存储器104可包括高速随机存储器,还可包括非易失性存储器,如一个或者多个磁性存储装置、闪存、或者其他非易失性固态存储器。在一些实例中,存储器104可进一步包括相对于处理器102远程设置的存储器,这些远程存储器可以通过网络连接至计算机终端10。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。

[0076] 传输装置106用于经由一个网络接收或者发送数据。上述的网络具体实例可包括计算机终端10的通信供应商提供的无线网络。在一个实例中,传输装置106包括一个网络适配器(Network Interface Controller, NIC),其可通过基站与其他网络设备相连从而可与互联网进行通讯。在一个实例中,传输装置106可以为射频(Radio Frequency, RF)模块,其用于通过无线方式与互联网进行通讯。

[0077] 本说明书中的各个实施方式均采用递进的方式描述,各个实施方式之间相同相似的部分互相参见即可,每个实施方式重点说明的都是与其他实施方式的不同之处。尤其,针对系统和设备的实施方式来说,均可以参照前述方法的实施方式的介绍对照解释。

[0078] 本领域内的技术人员应明白,本发明的实施方式可提供为方法、系统、或计算机程序产品。因此,本发明可采用完全硬件实施方式、完全软件实施方式、或结合软件和硬件方面的实施方式的形式。而且,本发明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。

[0079] 本发明是参照根据本发明实施方式的方法、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。

[0080] 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造品,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。

[0081] 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。

[0082] 在一个典型的配置中,计算设备包括一个或多个处理器(CPU)、输入/输出接口、网络接口和内存。

[0083] 内存可能包括计算机可读介质中的非永久性存储器,随机存取存储器(RAM)和/或非易失性内存等形式,如只读存储器(ROM)或闪存(flash RAM)。内存是计算机可读介质的示例。

[0084] 计算机可读介质包括永久性和非永久性、可移动和非可移动媒体可以由任何方法或技术来实现信息存储。信息可以是计算机可读指令、数据结构、程序的模块或其他数据。计算机的存储介质的例子包括,但不限于相变内存(PRAM)、静态随机存取存储器(SRAM)、动态随机存取存储器(DRAM)、其他类型的随机存取存储器(RAM)、只读存储器(ROM)、电可擦除可编程只读存储器(EEPROM)、快闪记忆体或其他内存技术、只读光盘只读存储器(CD-ROM)、数字多功能光盘(DVD)或其他光学存储、磁盒式磁带,磁带磁盘存储或其他磁性存储设备或任何其他非传输介质,可用于存储可以被计算设备访问的信息。按照本文中的界定,计算机可读介质不包括暂存电脑可读媒体(transitory media),如调制的数据信号和载波。

[0085] 还需要说明的是,术语“包括”、“包含”或者其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、商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、商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“包括一个……”限定的要素,并不排除在包括所述要素的过程、方法、商品或者设备中还存在另外的相同要素。

[0086] 以上所述仅为本申请的实施方式而已,并不用于限制本申请。对于本领域技术人员来说,本申请可以有各种更改和变化。凡在本申请的精神和原理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权利要求范围之内。

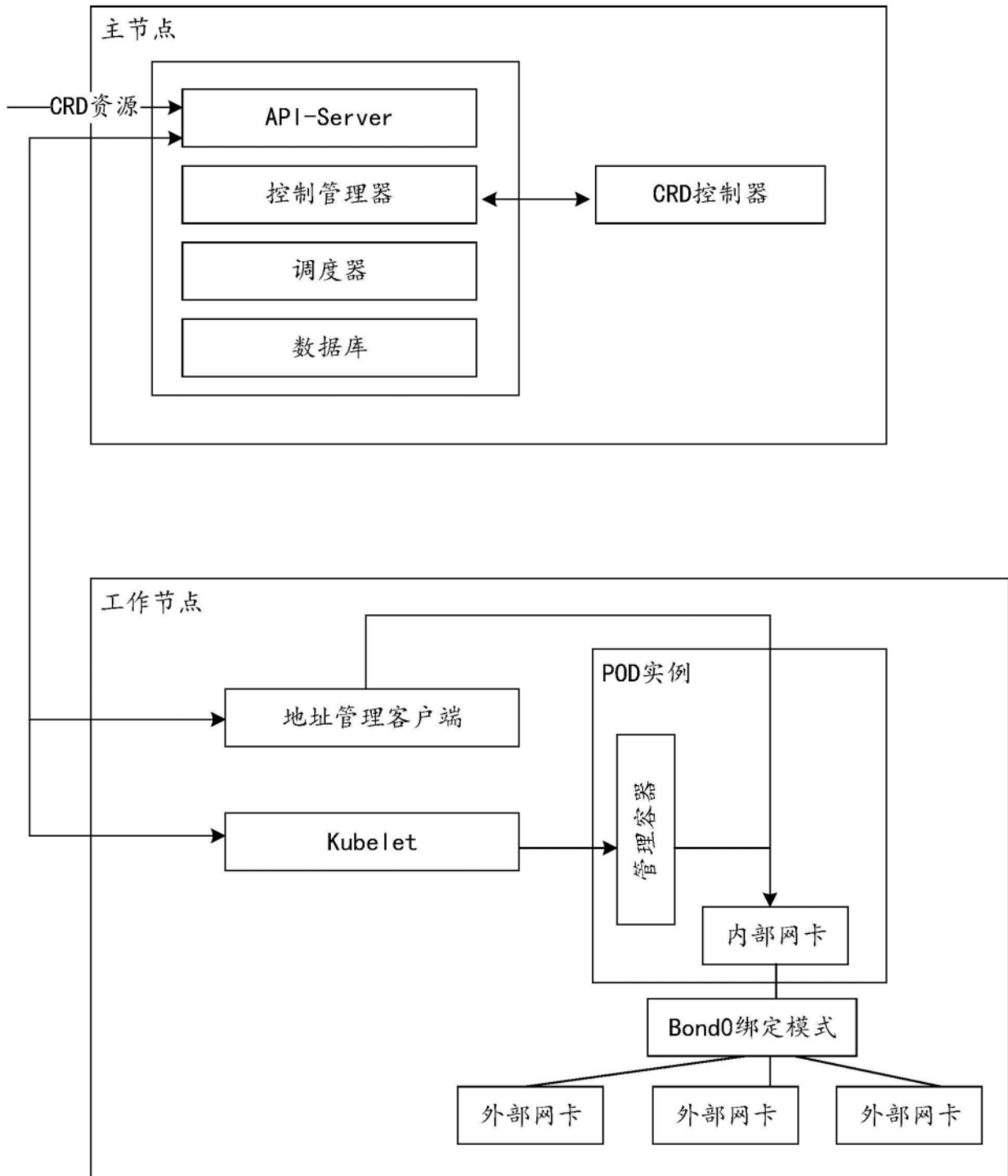


图1

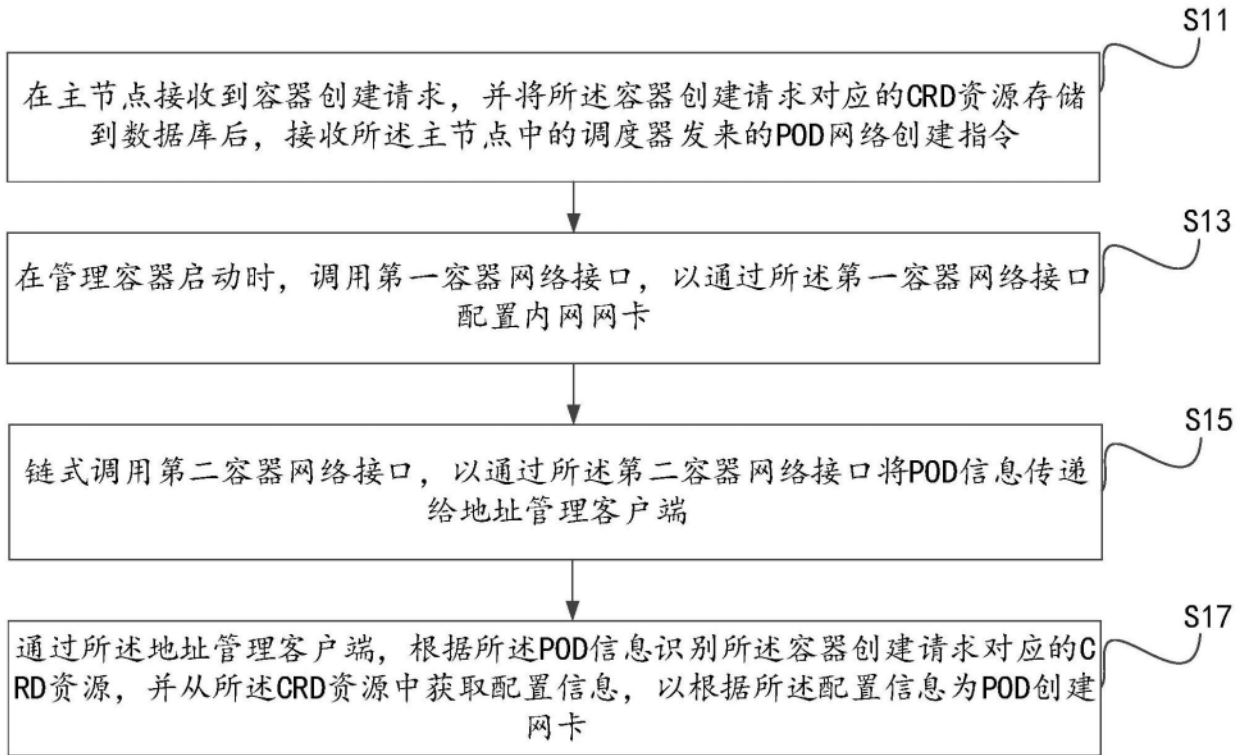


图2

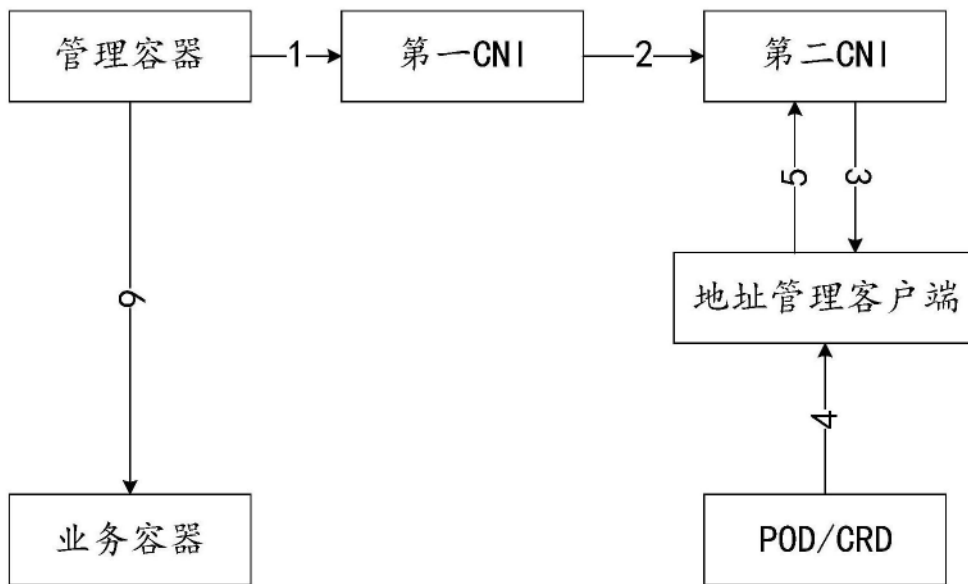


图3

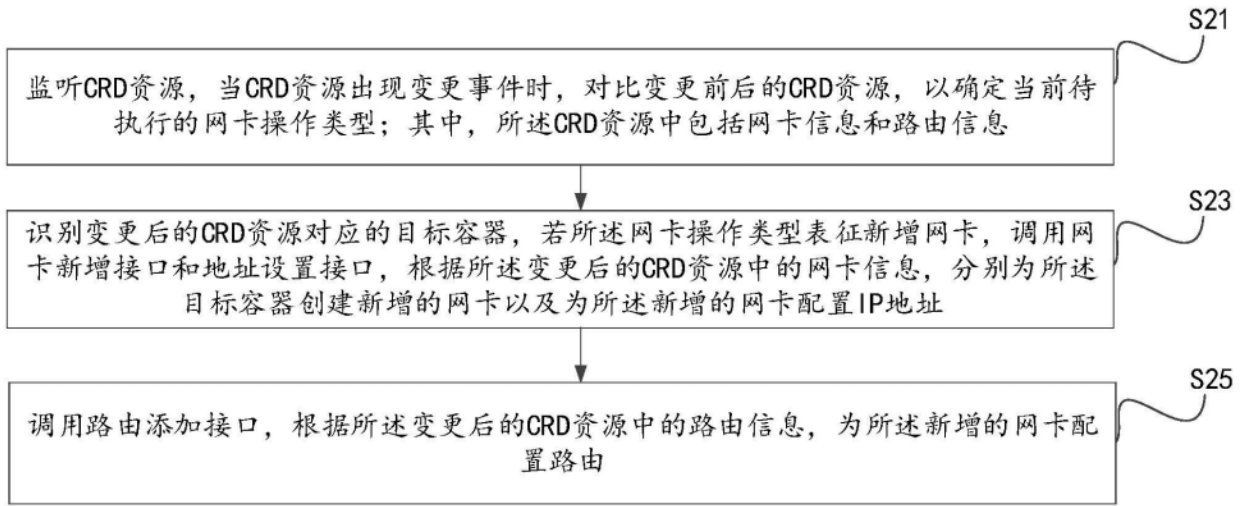


图4

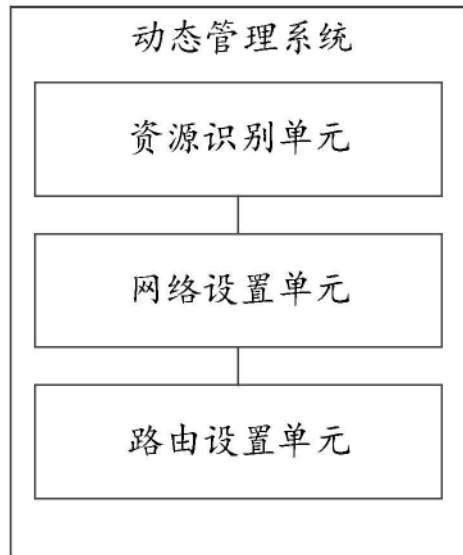


图5

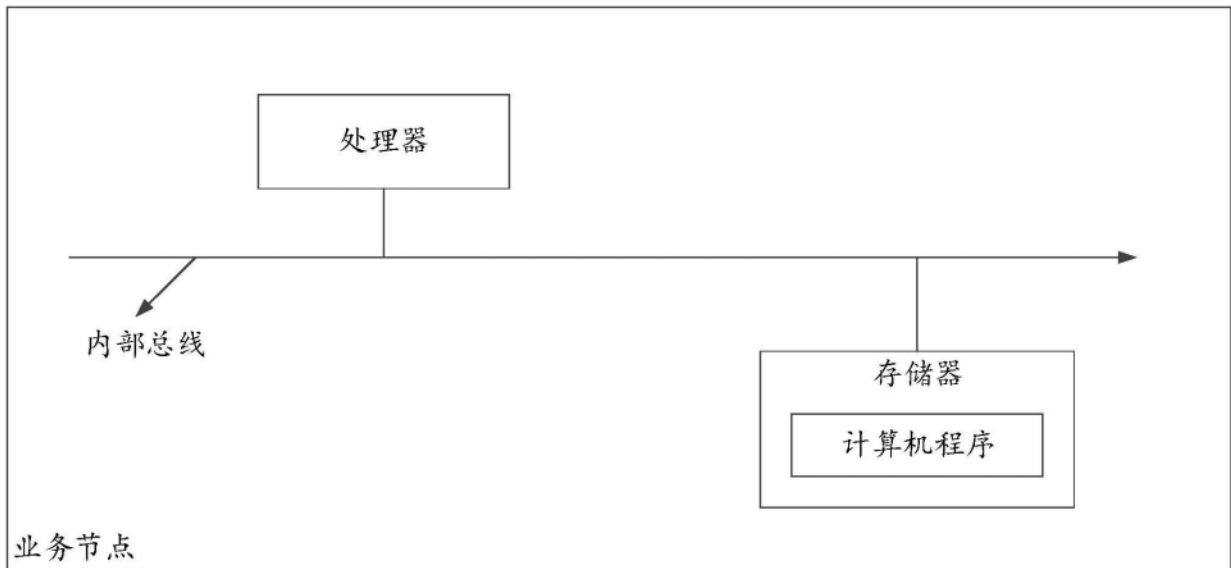


图6

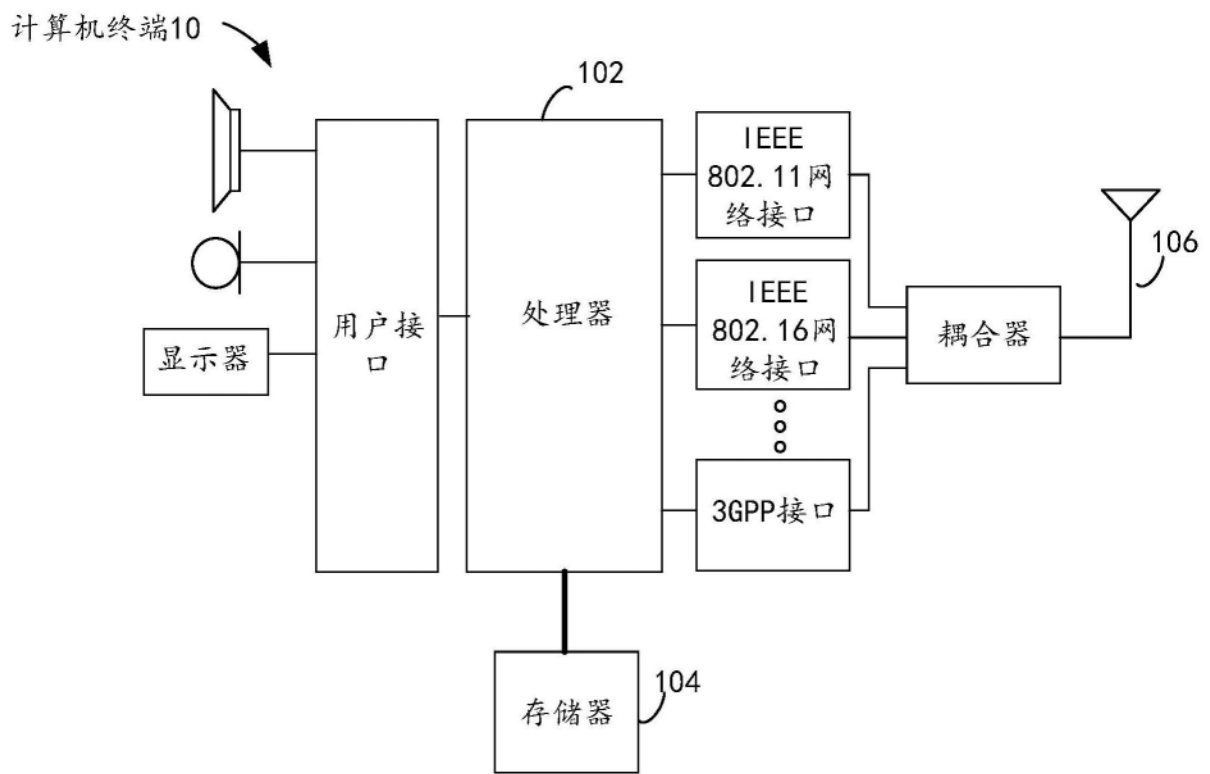


图7