



(21) 申请号 202010027431.2

(22) 申请日 2020.01.10

(65) 同一申请的已公布的文献号
申请公布号 CN 111277432 A

(43) 申请公布日 2020.06.12

(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 葛凯凯

(74) 专利代理机构 北京市立方律师事务所
11330
专利代理师 张筱宁 张海秀

(51) Int.Cl.
H04L 41/0813 (2022.01)
G06F 9/455 (2018.01)

(56) 对比文件

CN 109992279 A, 2019.07.09
CN 106878363 A, 2017.06.20
CN 103559063 A, 2014.02.05
CN 106506246 A, 2017.03.15
US 7519690 B1, 2009.04.14
CN 110244987 A, 2019.09.17
CN 108804618 A, 2018.11.13

审查员 刘琼艳

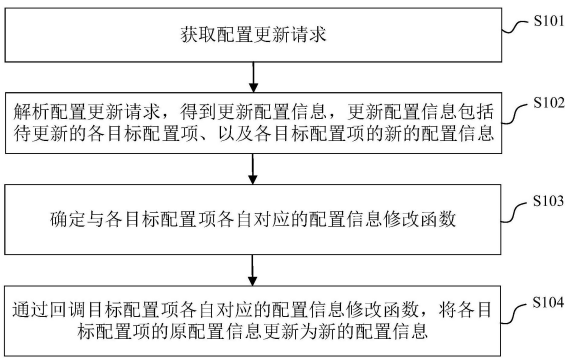
权利要求书4页 说明书17页 附图8页

(54) 发明名称

配置信息更新方法、装置、电子设备及存储
介质

(57) 摘要

本申请实施例提供了一种配置信息更新方法、装置、电子设备及存储介质,涉及信息处理技术领域。该方法包括:获取配置更新请求;解析配置更新请求,得到更新配置信息,更新配置信息包括待更新的各目标配置项的标识、以及各目标配置项的新的配置信息;确定与各目标配置项各自对应的配置信息修改函数;通过回调目标配置项各自对应的配置信息修改函数,将各目标配置项的当前配置信息更新为新的配置信息。本申请提供的技术方案,在对neutron服务组件进行配置更新时,不需要重启neutron服务组件,避免了由于重启造成的虚拟机网络瘫痪问题,不影响neutron服务组件的正常工作,提升了系统的稳定性。



1. 一种配置信息更新方法,其特征在于,所述方法的执行主体为neutron控制节点或neutron代理节点;其中,所述neutron控制节点或neutron代理节点中增加主线程之外的旁路线程;通过在所述旁路线程中运行的配置更新子系统执行所述方法,以更新对应节点的配置信息;

所述方法包括:

接收配置项设置请求,所述配置项设置请求包括各可更新配置项的标识、以及各所述可更新配置项各自所对应的配置信息修改函数;

根据所述配置项设置请求,建立各所述可更新配置项的标识、以及各所述可更新配置项各自所对应的配置信息修改函数之间的映射关系并存储;

获取配置更新请求;

解析所述配置更新请求,得到更新配置信息,所述更新配置信息包括待更新的各目标配置项的标识、以及各所述目标配置项的新的配置信息;

基于所述映射关系,以及各所述目标配置项的标识,确定与各所述目标配置项各自对应的配置信息修改函数,所述配置信息修改函数用于修改对应目标配置项的配置信息;

通过回调所述目标配置项各自对应的所述配置信息修改函数,将各所述目标配置项的当前配置信息更新为所述新的配置信息。

2. 根据权利要求1所述的方法,其特征在于,所述方法的执行主体为neutron代理节点,所述获取配置更新请求,包括以下至少一项:

接收neutron控制节点发送的配置更新请求,其中,所述配置更新请求为所述neutron控制节点通过接收到的第一表现层状态转化restful请求确定出的;

通过监测第一Unix套接字文件获取配置更新请求。

3. 根据权利要求2所述的方法,其特征在于,所述方法还包括:

接收第一命令行CLI配置更新请求,将接收到的配置更新请求写入所述第一Unix套接字文件。

4. 根据权利要求2或3所述的方法,其特征在于,所述neutron代理节点包括所述neutron控制节点的代理节点或neutron计算节点的代理节点。

5. 根据权利要求2所述的方法,其特征在于,所述第一restful请求中携带有所述neutron代理节点的节点标识,所述配置更新请求为所述neutron控制节点通过以下方式确定出的:

接收所述第一Restful请求;

解析所述第一Restful请求,基于解析结果得到各节点标识和每个节点标识所对应的配置更新请求。

6. 根据权利要求1所述的方法,其特征在于,所述方法的执行主体为neutron控制节点,所述获取配置更新请求包括以下至少一项:

接收第二Restful请求,基于所述第二Restful请求得到配置更新请求;

通过监测第二Unix套接字文件获取配置更新请求。

7. 根据权利要求6所述的方法,其特征在于,所述第二Restful请求中携带有所述neutron控制节点的节点标识,所述基于所述第二Restful请求得到配置更新请求,包括:

解析所述第二Restful请求,基于解析结果得到各节点标识和每个节点标识所对应的

配置更新请求；

将所述各节点标识中所述neutron控制节点的节点标识所对应的配置更新请求，确定为所述配置更新请求。

8. 根据权利要求1所述的方法，其特征在于，在接收到配置项设置请求之后，所述方法还包括：

根据所述配置项设置请求中的各可更新配置项的标识，获取对应的各可更新配置项的当前配置信息；

采用键值对形式的单例对象保存各可更新配置项的标识及对应的当前配置信息，其中，键值对中的键为可更新配置项的标识，值为可更新配置项的当前配置信息。

9. 根据权利要求1所述的方法，其特征在于，对于一目标配置项，所述配置信息修改函数包括以下任一项：

键值修改函数；

所述键值修改函数和配置项处理函数；

通过回调该目标配置项对应的所述配置信息修改函数，将该目标配置项的当前配置信息更新为所述新的配置信息，包括：

若所述配置信息修改函数为键值修改函数，则通过回调该目标配置项对应的所述键值修改函数，将所述目标配置项的单例对象的键值对中的值由当前配置信息修改为新的配置信息；

若所述配置信息修改函数为键值修改函数和配置项处理函数，则通过回调该目标配置项对应的所述键值修改函数，将所述目标配置项的单例对象的键值对中的值由当前配置信息修改为新的配置信息，并通过回调该目标配置项对应的所述配置项处理函数，将该目标配置项的当前配置信息更新为所述新的配置信息。

10. 一种配置信息更新方法，其特征在于，所述方法的执行主体为neutron控制节点，所述neutron控制节点中增加主线程之外的旁路线程；通过在旁路线程中运行的配置更新子系统执行所述方法，以更新所述neutron控制节点或neutron代理节点的配置信息；

所述方法包括：

接收配置项设置请求，所述配置项设置请求包括各可更新配置项的标识、以及各所述可更新配置项各自所对应的配置信息修改函数；

根据所述配置项设置请求，建立各所述可更新配置项的标识、以及各所述可更新配置项各自所对应的配置信息修改函数之间的映射关系并存储；

接收Restful请求，解析所述Restful请求，得到各节点标识和每个节点标识所对应的配置更新请求，其中，所述配置更新请求中包括更新配置信息，所述更新配置信息包括待更新的各目标配置项的标识、以及各所述目标配置项的新的配置信息；

若所述节点标识包括neutron控制节点的标识，则基于所述映射关系，以及各所述目标配置项的标识，确定与所述neutron控制节点的标识对应的各所述目标配置项各自对应的配置信息修改函数，并通过回调所述目标配置项各自对应的所述配置信息修改函数，将各所述目标配置项的当前配置信息更新为所述新的配置信息，所述配置信息修改函数用于修改对应目标配置项的配置信息；

若所述节点标识包括neutron代理节点的标识，则将所述配置更新请求发送至对应的

neutron代理节点,以使所述neutron代理节点根据接收到的配置更新请求,基于所述映射关系,以及各所述目标配置项的标识,确定该配置更新请求中的各所述目标配置项各自对应的配置信息修改函数,并通过回调所述目标配置项各自对应的所述配置信息修改函数,将各所述目标配置项的当前配置信息更新为所述新的配置信息。

11.一种配置信息更新装置,其特征在于,所述装置的执行主体为neutron控制节点或neutron代理节点;其中,所述neutron控制节点或neutron代理节点中增加主线程之外的旁路线程;所述装置应用在旁路线程中运行的配置更新子系统中,以更新对应节点的配置信息;

所述装置包括:

配置项设置模块,用于接收配置项设置请求,所述配置项设置请求包括各可更新配置项的标识、以及各所述可更新配置项各自所对应的配置信息修改函数;根据所述配置项设置请求,建立各所述可更新配置项的标识、以及各所述可更新配置项各自所对应的配置信息修改函数之间的映射关系并存储;

获取模块,用于获取配置更新请求;

解析模块,用于解析所述配置更新请求,得到更新配置信息,所述更新配置信息包括待更新的各目标配置项的标识、以及各所述目标配置项的新的配置信息;

确定模块,用于基于所述映射关系,以及各所述目标配置项的标识,确定与各所述目标配置项各自对应的配置信息修改函数,所述配置信息修改函数用于修改对应目标配置项的配置信息;

更新模块,用于通过回调所述目标配置项各自对应的所述配置信息修改函数,将各所述目标配置项的当前配置信息更新为所述新的配置信息。

12.一种配置信息更新装置,其特征在于,所述装置的执行主体为neutron控制节点,所述neutron控制节点中增加主线程之外的旁路线程;所述装置应用在旁路线程中运行的配置更新子系统中,以更新所述neutron控制节点或neutron代理节点的配置信息;

所述装置包括:

配置项设置模块,用于接收配置项设置请求,所述配置项设置请求包括各可更新配置项的标识、以及各所述可更新配置项各自所对应的配置信息修改函数;根据所述配置项设置请求,建立各所述可更新配置项的标识、以及各所述可更新配置项各自所对应的配置信息修改函数之间的映射关系并存储;

接收模块,用于接收Restful请求,解析所述Restful请求,得到各节点标识和每个节点标识所对应的配置更新请求,其中,所述配置更新请求中包括更新配置信息,所述更新配置信息包括待更新的各目标配置项的标识、以及各所述目标配置项的新的配置信息,所述配置信息修改函数用于修改对应目标配置项的配置信息;

确定模块,用于若所述节点标识包括neutron控制节点的标识,则基于所述映射关系,以及各所述目标配置项的标识,确定与所述neutron控制节点的标识对应的各所述目标配置项各自对应的配置信息修改函数,并通过回调所述目标配置项各自对应的所述配置信息修改函数,将各所述目标配置项的当前配置信息更新为所述新的配置信息;

发送模块,用于若所述节点标识包括neutron代理节点的标识,则将所述配置更新请求发送至对应的neutron代理节点,以使所述neutron代理节点根据接收到的配置更新请求,

基于所述映射关系,以及各所述目标配置项的标识,确定该配置更新请求中的各所述目标配置项各自对应的配置信息修改函数,并通过回调所述目标配置项各自对应的所述配置信息修改函数,将各所述目标配置项的当前配置信息更新为所述新的配置信息。

13.一种电子设备,其特征在于,所述电子设备包括:

一个或多个处理器;

存储器;

一个或多个计算机程序,其中所述一个或多个计算机程序被存储在所述存储器中并被配置为由所述一个或多个处理器执行,所述一个或多个程序被配置用于执行根据权利要求1~10任一项所述的方法。

14.一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用于存储计算机程序,当所述计算机程序在处理器上运行时,使得所述处理器可以执行权利要求1~10任一项所述的方法。

配置信息更新方法、装置、电子设备及存储介质

技术领域

[0001] 本申请涉及计算机技术领域,具体而言,本申请涉及一种配置信息更新方法、装置、电子设备及存储介质。

背景技术

[0002] 随着云计算的发展,大量企业都计划或正在使用云计算,云计算模式包括公有云和私有云。公有云通常指第三方提供商用户能够使用的云,公有云一般可通过Internet使用,可能是免费或成本低廉的。私有云是为一个客户单独使用而构建的,能够提供对数据安全性和服务质量的最有效控制。

[0003] 无论是在私有云还是在公有云环境中,网络资源都是不可或缺的,比如虚拟网卡、虚拟路由器、虚拟防火墙、虚拟交换机等,这些网络资源功能的实现都依赖于网络流表,这些网络流表提供了数据的转发依据,Openstack中的neutron服务也依赖网络流表。而网络流表的缺失或者异常,可能会导致虚拟机网络通信等功能的故障。目前已经上线的neutron服务如果要更新调整配置,需要修改完配置文件之后重启服务,而neutron重启服务会触发网络流表的清空与重建,在一些大规模的云环境中可能会导致虚拟机网络瘫痪,给用户服务带来巨大的影响。

发明内容

[0004] 本申请提供了一种配置信息更新方法、装置、电子设备及存储介质,可以解决现有技术中存在的问题。

[0005] 本申请实施例提供的具体技术方案如下:

[0006] 第一方面,本申请实施例提供了一种配置信息更新方法,该方法包括:

[0007] 获取配置更新请求;

[0008] 解析配置更新请求,得到更新配置信息,更新配置信息包括待更新的各目标配置项的标识、以及各目标配置项的新的配置信息;

[0009] 确定与各目标配置项各自对应的配置信息修改函数;

[0010] 通过回调目标配置项各自对应的配置信息修改函数,将各目标配置项的当前配置信息更新为新的配置信息。

[0011] 在一种可能的实现方式中,方法的执行主体为neutron代理节点,获取配置更新请求,包括以下至少一项:

[0012] 接收neutron控制节点发送的配置更新请求,其中,配置更新请求为neutron控制节点通过接收到的第一表现层状态转化restful请求确定出的;

[0013] 通过监测第一Unix套接字文件获取配置更新请求。

[0014] 在一种可能的实现方式中,方法还包括:

[0015] 接收第一命令行CLI配置更新请求,将接收到的配置更新请求写入第一Unix套接字文件。

[0016] 在一种可能的实现方式中,neutron代理节点包括neutron控制节点的代理节点或neutron计算节点的代理节点。

[0017] 在一种可能的实现方式中,第一restful请求中携带有neutron代理节点的节点标识,配置更新请求为neutron控制节点通过以下方式确定出的:

[0018] 接收第一Restful请求;

[0019] 解析第一Restful请求,基于解析结果得到各节点标识和每个节点标识所对应的配置更新请求。

[0020] 在一种可能的实现方式中,方法的执行主体为neutron控制节点,获取配置更新请求包括以下至少一项:

[0021] 接收第二Restful请求,基于第二Restful请求得到配置更新请求;

[0022] 通过监测第二Unix套接字文件获取配置更新请求。

[0023] 在一种可能的实现方式中,方法还包括:

[0024] 接收第二CLI配置更新请求,将接收到的配置更新请求写入第二Unix套接字文件。

[0025] 在一种可能的实现方式中,第二Restful请求中携带有neutron控制节点的节点标识,基于第二Restful请求得到配置更新请求,包括:

[0026] 解析第二Restful请求,基于解析结果得到各节点标识和每个节点标识所对应的配置更新请求;

[0027] 将各节点标识中neutron控制节点的节点标识所对应的配置更新请求,确定为配置更新请求。

[0028] 在一种可能的实现方式中,在获取配置更新请求之前,方法还包括:

[0029] 接收配置项设置请求,配置项设置请求包括各可更新配置项的标识、以及各可更新配置项各自所对应的配置信息修改函数;

[0030] 根据配置项设置请求,建立各可更新配置项的标识、以及各可更新配置项各自所对应的配置信息修改函数之间的映射关系并存储;

[0031] 确定与各目标配置项各自对应的配置信息修改函数,包括:

[0032] 基于映射关系,以及各目标配置项的标识,确定与各目标配置项各自对应的配置信息修改函数。

[0033] 在一种可能的实现方式中,在接收到配置项设置请求之后,方法还包括:

[0034] 根据配置项设置请求中的各可更新配置项的标识,获取对应的各可更新配置项的当前配置信息;

[0035] 采用键值对形式的单例对象保存各可更新配置项的标识及对应的当前配置信息,其中,键值对中的键为可更新配置项的标识,值为可更新配置项的当前配置信息。

[0036] 在一种可能的实现方式中,对于一目标配置项,配置信息修改函数包括以下任一项:

[0037] 键值修改函数;

[0038] 键值修改函数和配置项处理函数;

[0039] 通过回调该目标配置项对应的配置信息修改函数,将该目标配置项的当前配置信息更新为新的配置信息,包括:

[0040] 若配置信息修改函数为键值修改函数,则通过回调该目标配置项对应的键值修改

函数,将目标配置项的单例对象的键值对中的值由当前配置信息修改为新的配置信息;

[0041] 若配置信息修改函数为键值修改函数和配置项处理函数,则通过回调该目标配置项对应的键值修改函数,将目标配置项的单例对象的键值对中的值由当前配置信息修改为新的配置信息,并通过回调该目标配置项对应的配置项处理函数,将该目标配置项的当前配置信息更新为新的配置信息。

[0042] 第二方面,本申请实施例提供了一种配置信息更新方法,该方法包括:

[0043] 接收Restful请求,解析Restful请求,得到各节点标识和每个节点标识所对应的配置更新请求,其中,配置更新请求中包括更新配置信息,更新配置信息包括待更新的各目标配置项的标识、以及各目标配置项的新的配置信息;

[0044] 若节点标识包括neutron控制节点的标识,则确定与neutron控制节点的标识对应的各目标配置项各自对应的配置信息修改函数,并通过回调目标配置项各自对应的配置信息修改函数,将各目标配置项的当前配置信息更新为新的配置信息;

[0045] 若节点标识包括neutron代理节点的标识,则将配置更新请求发送至对应的neutron代理节点,以使neutron代理节点根据接收到的配置更新请求,确定该配置更新请求中的各目标配置项各自对应的配置信息修改函数,并通过回调目标配置项各自对应的配置信息修改函数,将各目标配置项的当前配置信息更新为新的配置信息。

[0046] 第三方面,本发明实施例提供了一种配置信息更新装置,该装置包括:

[0047] 获取模块,用于获取配置更新请求;

[0048] 解析模块,用于解析配置更新请求,得到更新配置信息,更新配置信息包括待更新的各目标配置项的标识、以及各目标配置项的新的配置信息;

[0049] 确定模块,用于确定与各目标配置项各自对应的配置信息修改函数;

[0050] 更新模块,用于通过回调目标配置项各自对应的配置信息修改函数,将各目标配置项的当前配置信息更新为新的配置信息。

[0051] 第四方面,本发明实施例提供了一种配置信息更新装置,该装置包括:

[0052] 接收模块,用于接收Restful请求,解析Restful请求,得到各节点标识和每个节点标识所对应的配置更新请求,其中,配置更新请求中包括更新配置信息,更新配置信息包括待更新的各目标配置项的标识、以及各目标配置项的新的配置信息;

[0053] 确定模块,用于若节点标识包括neutron控制节点的标识,则确定与neutron控制节点的标识对应的各目标配置项各自对应的配置信息修改函数,并通过回调目标配置项各自对应的配置信息修改函数,将各目标配置项的当前配置信息更新为新的配置信息;

[0054] 发送模块,用于若节点标识包括neutron代理节点的标识,则将配置更新请求发送至对应的neutron代理节点,以使neutron代理节点根据接收到的配置更新请求,确定该配置更新请求中的各目标配置项各自对应的配置信息修改函数,并通过回调目标配置项各自对应的配置信息修改函数,将各目标配置项的当前配置信息更新为新的配置信息。

[0055] 第五方面,本发明实施例还提供了一种电子设备,该电子设备包括一个或多个处理器;存储器;一个或多个计算机程序,其中一个或多个计算机程序被存储在存储器中并被配置为由一个或多个处理器执行,一个或多个程序被配置用于执行如本申请的第一方面或第二方面所示的方法。

[0056] 第六方面,本发明实施例还提供了一种计算机可读存储介质,计算机可读存储介

质用于存储计算机程序,当计算机程序在处理器上运行时,使得处理器可以执行如本申请的第一方面或第二方面所示的方法。

[0057] 本申请提供的技术方案带来的有益效果是:

[0058] 本申请提供了一种配置信息更新方法、装置、电子设备及存储介质,获取配置更新请求;解析配置更新请求,得到更新配置信息,更新配置信息包括待更新的各目标配置项的标识以及各目标配置项的新的配置信息;确定与各目标配置项各自对应的配置信息修改函数;通过回调目标配置项各自对应的配置信息修改函数,将各目标配置项的当前配置信息更新为新的配置信息。本申请提供的技术方案,在对neutron服务组件进行配置更新时,通过回调目标配置项各自对应的配置信息修改函数,来实现将当前配置信息更新为新的配置信息,不需要重启neutron服务组件,避免了由于重启造成的虚拟机网络瘫痪问题,不影响neutron服务组件的正常工作,提升了系统的稳定性。

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配置信息更新方法的流程图;

[0061] 图2a-2b为本申请实施例提供的控制节点、计算节点的结构示意图;

[0062] 图3为本申请实施例提供的控制节点通过restful请求确定配置更新请求的示意图;

[0063] 图4为本申请实施例提供的通过监测套接字文件确定配置更新请求的示意图;

[0064] 图5为本申请实施例提供的配置信息注册过程示意图;

[0065] 图6为本申请实施例提供的配置更新子系统的结构示意图;

[0066] 图7为本申请实施例提供的配置信息更新方法的流程图;

[0067] 图8为本申请实施例提供的配置信息更新的具体过程的流程图;

[0068] 图9为本申请实施例提供的配置信息更新方法的系统架构图;

[0069] 图10为本申请实施例提供的配置信息更新装置的结构示意图;

[0070] 图11为本申请实施例提供的配置信息更新装置的结构示意图;

[0071] 图12为本申请实施例提供的一种电子设备的结构示意图。

具体实施方式

[0072] 下面详细描述本申请的实施例,所述实施例的示例在附图中示出,其中自始至终相同或类似的标号表示相同或类似的元件或具有相同或类似功能的元件。下面通过参考附图描述的实施例是示例性的,仅用于解释本申请,而不能解释为对本申请的限制。

[0073] 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”、“所述”和“该”也可包括复数形式。应该进一步理解的是,本申请的说明书中使用的措辞“包括”是指存在所述特征、整数、步骤、操作、元件和/或组件,但是并不排除存在或添加一个或多个其他特征、整数、步骤、操作、元件、组件和/或它们的组。应该理解,当我们称元件被“连接”或“耦接”到另一元件时,它可以直接连接或耦接到其他元件,或者也可以存在中间元件。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。这里使用的措

辞“和/或”包括一个或多个相关联的列出项的全部或任一单元和全部组合。

[0074] 下面以具体地实施例对本申请的技术方案以及本申请的技术方案如何解决上述技术问题进行详细说明。下面这几个具体的实施例可以相互结合,对于相同或相似的概念或过程可能在某些实施例中不再赘述。下面将结合附图,对本申请的实施例进行描述。

[0075] 名词解释:

[0076] 基础设施即服务 (Infrastructure as a Service, IaaS) 是指将IT基础设施作为一种服务通过网络对外提供,并根据用户对资源的实际使用量或占用量进行计费的一种服务模式。

[0077] OpenStack: 开源的云计算管理平台,通过命令或者基于Web的可视化控制面板来管理IaaS云端的资源池(服务器、存储和网络)。

[0078] neutron: Openstack中的网络组件,用来管理各种网络资源的生命周期。

[0079] 本申请技术方案可以应用于neutron服务组件的配置信息的更新,neutron服务组件包括至少一个neutron控制节点(下文简称控制节点)和至少一个neutron计算节点(下文简称计算节点),控制节点包括至少一个neutron代理节点,计算节点包括至少一个neutron代理节点。本申请技术方案通过在控制节点的neutron代理节点(下文简称控制节点的代理节点)、以及计算节点的neutron代理节点(下文简称计算节点的代理节点)中增加旁路线程,在旁路线程中运行配置更新子系统的方式实现neutron服务组件的更新,更新时不需要重启,不影响neutron服务组件的主线程正常工作。

[0080] 本申请实施例提供了一种配置信息更新方法,如图1所示,该方法包括:

[0081] 步骤S101, 获取配置更新请求;

[0082] 步骤S102, 解析配置更新请求, 得到更新配置信息, 更新配置信息包括待更新的各目标配置项的标识、以及各目标配置项的新的配置信息;

[0083] 其中, 配置更新请求可以是针对neutron服务组件中的控制节点或者计算节点中的配置信息进行更新的请求。配置更新请求中携带有待更新的各目标配置项的标识、以及各目标配置项的新的配置信息。

[0084] 将接收到的配置更新请求进行解析, 得到至少一个待更新的目标配置项的标识、以及每个目标配置项的新的配置信息, 其中, 目标配置项为待更新的控制节点或者计算节点中的配置项, 目标配置项的新的配置信息为更新之后的目标配置项的配置信息, 目标配置项的标识具体为配置项的标识, 可以为配置项的名称、也可以为key, 通常为key。

[0085] 获取配置更新请求的具体方式可以通过以下实施例来实现。

[0086] 在一种可能的实现方式中, 该方法的执行主体为neutron代理节点, 获取配置更新请求, 包括以下至少一项:

[0087] 接收控制节点发送的配置更新请求, 其中, 配置更新请求为控制节点通过接收到的第一表现层状态转化 (Representational State Transfer, REST) restful请求确定出的;

[0088] 通过监测第一Unix套接字文件获取配置更新请求。

[0089] 在实际应用中, 配置更新请求可以是利用超文本传输协议 (Hyper Text Transfer Protocol, HTTP) 网络传输, 通过restful应用程序接口 (Application Programming Interface, API) 接收到的请求确定出的, 当执行主体为neutron代理节点时, 控制节点通过

server进程接收第一restful请求,根据第一restful请求确定出配置更新请求,该配置更新请求有可能是针对控制节点的server进程的更新请求,也有可能是针对neutron代理节点的更新请求,当控制节点确定出的配置更新请求是针对neutron代理节点时,将配置更新请求发送到对应的neutron代理节点,该neutron代理节点接收到配置更新请求,后续对该配置更新请求进行处理;配置更新请求还有可能是用户通过输入命令的形式直接输入的命令行请求,无论是控制节点还是neutron代理节点,都可以新建一个线程,在该线程中通过监测各自对应的Unix套接字文件的形式获取到命令行形式的配置更新请求。

[0090] 本申请实施例中,通过restful请求的方式获取到配置更新请求方式,可以适用于私有云的运维系统,供运维管理人员使用,不仅如此,接收restful请求的客户端还可以提供给任意的第三方调用,比如配置中心。而通过命令行形式输入配置更新请求,通过监测Unix套接字文件获取配置更新请求的方式,可以通过后台服务提供给开发人员使用。本实施例提供了两种获取配置更新请求的方式,使配置更新使用起来更加的方便,适合满足更多的需求。

[0091] 另外,第一Unix套接字文件中的配置更新请求,是通过以下方式得到的:

[0092] 在一种可能的实现方式中,该方法还包括:

[0093] 接收第一命令行界面(command-line interface,CLI)配置更新请求,将接收到的配置更新请求写入第一Unix套接字文件。

[0094] 在实际应用中,neutron代理节点接收到命令行形式的配置更新请求,将该配置更新请求写入Unix套接字文件。Unix套接字文件为预配置的neutron代理节点的本地文件,通过Unix套接字文件能够实现不同进程之间的通信。Unix套接字仅仅进行数据复制,不会执行在网络协议栈中需要处理的添加、删除报文头、计算校验和、计算报文顺序等复杂操作,因此,通过使用Unix套接字进行同一个计算机的不同进程之间的通信,不仅操作简单而且高效。

[0095] 其中,neutron代理节点也可以是neutron服务组件中的不同节点的代理节点。

[0096] 在一种可能的实现方式中,neutron代理节点可以为控制节点的代理节点或计算节点的代理节点。

[0097] 在实际应用中,neutron代理节点可以是控制节点的代理节点,代理控制节点实现相应的功能;neutron代理节点还可以是计算节点的代理节点,代理计算节点实现相应的功能。

[0098] 在一示例中,如图2a所示,控制节点中包括控制节点的主服务节点(图中所示的neutron-server,也可以理解为服务进程)和多个代理节点,neutron-server用于对外提供OpenStack网络API、接收网络请求、调用plugin处理请求等,例如,neutron-server承担着接收restful请求并分发处理的任务。控制节点的代理节点可以包括neutron-dhcp-agent、neutron-l3-agent、neutron-metadata-agent、neutron-openvswitch-agent。这些代理节点的功能如下:neutron-dhcp-agent用于管理neutron网络的动态主机配置协议(Dynamic Host Configuration Protocol,DHCP)服务;neutron-l3-agent用于管理neutron虚拟路由器服务;neutron-metadata-agent用于访问虚拟机的元数据服务;neutron-openvswitch-agent用于管理neutron的虚拟交换机。

[0099] Neutron代理节点还可以是计算节点的代理节点,代理计算节点的实现相应的功

能。如图2b所示,计算节点的代理节点可以包括neutron-openvswitch-agent,用于管理neutron的虚拟交换机。其中,控制节点的代理节点、以及计算节点中的neutron代理节点中分别运行有预配置的配置更新子系统,用于实现各节点的配置信息的更新。

[0100] 确定代理节点对应的配置更新请求的具体方式,可以通过以下实施例来实现。

[0101] 在一种可能的实现方式中,第一restful请求中携带有neutron代理节点的节点标识,配置更新请求为控制节点通过以下方式确定出的:

[0102] 接收第一Restful请求;

[0103] 解析第一Restful请求,基于解析结果得到各节点标识和每个节点标识所对应的配置更新请求。

[0104] 在实际应用中,控制节点通过neutron server进程接收到restful请求,并对第一restful请求进行解析,基于解析结果可以得到各节点标识,以及每个节点标识所对应的配置更新请求。其中,配置更新请求可以是针对控制节点的server进程的,也可以是针对控制节点的代理节点或计算节点的代理节点的,如果是针对控制节点的server进程的,则控制节点进行相应处理,如果是针对控制节点或者计算节点的代理节点的,则控制节点可以根据节点标识将配置更新请求发送至对应的代理节点。具体的,若控制节点接收到的该Restful请求中携带有neutron代理节点的节点标识,则表示该Restful请求中包含有针对该节点标识所对应的neutron代理节点的配置更新请求,控制节点则可以将对应的配置更新请求发送至相应的neutron代理节点,以使neutron代理节点基于接收到的配置更新请求进行配置信息的更新。

[0105] 上述本申请的可选实施例是以neutron代理节点为执行主体的实现方式,本申请中的方法还可以以控制节点为执行主体,具体实现方式见以下实施例。

[0106] 在一种可能的实现方式中,该方法的执行主体为控制节点,获取配置更新请求包括以下至少一项:

[0107] 接收第二Restful请求,基于第二Restful请求得到配置更新请求;

[0108] 通过监测第二Unix套接字文件获取配置更新请求。

[0109] 在实际应用中,控制节点可以通过接收到的第二Restful请求确定出配置更新请求,具体的,控制节点的server进程通过HTTP网络接收第二Restful请求,根据第二restful请求确定出配置更新请求。另外,配置更新请求还有可能是用户通过输入命令的形式直接输入的命令行请求,无论是控制节点还是neutron代理节点,都可以新建一个线程,在该线程中通过监测各自对应的Unix套接字文件的形式获取到命令行形式的配置更新请求。

[0110] 本申请实施例中,通过restful请求的方式获取到配置更新请求方式,可以适用于私有云的运维系统,供运维管理人员使用,不仅如此,接收restful请求的客户端还可以提供给任意的第三方调用,比如配置中心。而通过命令行形式输入配置更新请求,通过监测Unix套接字文件获取配置更新请求的方式,可以通过后台服务提供给开发人员使用。本实施例提供了两种获取配置更新请求的方式,使配置更新使用起来更加的方便,适合满足更多的需求。

[0111] 另外,第二Unix套接字文件中的配置更新请求,是通过以下方式得到的:

[0112] 在一种可能的实现方式中,该方法还包括:

[0113] 接收第二CLI配置更新请求,将接收到的配置更新请求写入第二Unix套接字文件。

[0114] 在实际应用中,控制节点接收到命令行形式的配置更新请求,将该配置更新请求写入Unix套接字文件。Unix套接字文件为本地文件,通过Unix套接字文件能够实现不同进程之间的通信。Unix套接字仅仅进行数据复制,不会执行在网络协议栈中需要处理的添加、删除报文头、计算校验和、计算报文顺序等复杂操作,因此,通过使用Unix套接字进行同一个计算机的不同进程之间的通信,不仅操作简单而且高效。

[0115] 确定控制节点对应的配置更新请求的具体方式,可以通过以下实施例来实现。

[0116] 在一种可能的实现方式中,第二Restful请求中携带有控制节点的节点标识,基于第二Restful请求得到配置更新请求,包括:

[0117] 解析第二Restful请求,基于解析结果得到各节点标识和每个节点标识所对应的配置更新请求;

[0118] 将各节点标识中控制节点的节点标识所对应的配置更新请求,确定为配置更新请求。

[0119] 在实际应用中,控制节点接收到的第二Restful请求中携带有控制节点的节点标识,控制节点通过server进程接收第二restful请求,并对第二restful请求进行解析,基于解析结果可以得到各节点标识,以及每个节点标识所对应的配置更新请求。将各节点标识中控制节点的节点标识所对应的配置更新请求,确定为配置更新请求。其中,控制节点的节点标识对应的配置更新请求是针对控制节点的server进程的配置更新请求。

[0120] 下面通过一个具体的实施例来说明通过restful请求接收配置更新请求的实现过程。

[0121] 在一示例中,控制节点通过restful请求确定配置更新请求的示意图如图3所示。控制节点在server进程(图中所示的neutron-server)中增加动态修改配置的API,server进程中的Web server通过该API接收restful请求,对该restful请求进行解析之后,得到各节点标识和每个节点标识所对应的配置更新请求,根据节点标识确定配置更新请求的发送路径,将该路径传递给路由映射,然后配置对应的路由处理函数,如果配置更新请求是针对控制节点的server进程的,则直接发送到控制节点自己的配置更新子系统进行处理,如果配置更新请求是针对控制节点的代理节点或者计算节点的代理节点的,则通过远程过程调用(Remote Procedure Call,RPC)通过高级消息队列协议的服务器(图中所示的Rabbitmq)将配置更新请求发送到相应的代理节点(图中所示的neutron-openvswitch-agent、neutron-dhcp-agent)。

[0122] 下面通过一个具体的实施例来说明通过命令行形式的请求确定配置更新请求的实现过程。

[0123] 在一示例中,通过监测套接字文件确定配置更新请求的示意图如图4所示,命令行界面客户端(图中所示的Cli client)接收用户输入的命令行请求,图中所示的neutron服务包括:控制节点中的server进程、控制节点的代理节点以及计算节点的代理节点,每个节点中均可以包括Unix套接字监测模块(如图中所示的Unix套接字监测)和配置更新子系统。接收到命令行请求之后,写入一个本地的Unix套接字Sock文件,各个节点通过监测Sock文件的形式获取到命令行形式的配置更新请求,并将该配置行请求发送至配置更新子系统进行处理。

[0124] 步骤S103,确定与各目标配置项各自对应的配置信息修改函数;

[0125] 其中,配置信息修改函数是用于修改各配置项的配置信息的函数,配置信息修改函数可以是修改配置文件中的配置项的字段值的一段程序语句,也可以是创建新的线程的函数,或者关闭线程的函数等,本申请对此不做限定。

[0126] 在获取到各个节点对应的配置信息更新请求之前,需要对各个节点中的配置项以及配置项相关的信息进行注册,注册之后才能根据配置信息更新请求进行相应节点的更新。

[0127] 在一种可能的实现方式中,在获取配置更新请求之前,该方法还包括:

[0128] 接收配置项设置请求,配置项设置请求包括各可更新配置项的标识、以及各可更新配置项各自所对应的配置信息修改函数;

[0129] 根据配置项设置请求,建立各可更新配置项的标识、以及各可更新配置项各自所对应的配置信息修改函数之间的映射关系并存储;

[0130] 确定与各目标配置项各自对应的配置信息修改函数,包括:

[0131] 基于映射关系,以及各目标配置项的标识,确定与各目标配置项各自对应的配置信息修改函数。

[0132] 本实施例中的获取配置更新请求之前是相对于上述接收到的配置更新请求而言的,可以在各个节点启动时,将各个节点相关的配置信息在配置更新子系统中进行注册,通过对配置项以及对应的配置信息修改函数的设置来实现配置信息的注册。接收配置项设置请求,该请求中包括至少一个可更新的配置项的标识,以及每个可更新配置项对应的配置信息修改函数,建立各个可更新配置项的标识和对应的配置信息修改函数的映射关系,并存储,经过注册之后,当接收到配置更新请求时,配置更新子系统才能进行响应,基于各可更新配置项的标识、以及各可更新配置项各自所对应的配置信息修改函数之间的映射关系,确定与目标配置项的标识对应的配置信息修改函数,对可更新配置项的配置信息进行相应的更新。

[0133] 在一种可能的实现方式中,在接收到配置项设置请求之后,该方法还包括:

[0134] 根据配置项设置请求中的各可更新配置项的标识,获取对应的各可更新配置项的当前配置信息;

[0135] 采用键值对形式的单例对象保存各可更新配置项的标识及对应的当前配置信息,其中,键值对中的键为可更新配置项的标识,值为可更新配置项的当前配置信息。

[0136] 在实际应用中,在接收到配置项设置请求之后,根据各可更新配置项的标识,能够获取的各可更新配置项的当前配置信息,采用单例对象全局保存各可更新配置项的标识和对应的当前配置信息,该单例对象可以是键值对的形式,例如,一般程序的配置项都是类似“key=value”的键值对的形式,键值对中的键为可更新配置项的标识,值为可更新配置项的当前配置信息,比如“pool_size=5”,表示一个线程池里面线程的个数是5,其中,pool_size为键,也就是可更新配置项的key,5为值,也就是该各更新配置对象的当前配置信息。注册配置项的时候,注册的就是这个key的值,把每个节点相关联的配置项的key的值注册给配置更新子系统。

[0137] 在进行注册时,通过单例对象在内存中全局保存配置项,例如,单例对象可以通过字段或者散列函数hash保存配置项对应的“key=value”键值对。本申请实施例中,采用单例对象保存配置项能够避免配置项的重复创建,减少重复创建配置项造成的系统开销,节

省内存。

[0138] 由于neutron组件的各个节点的功能不一样,因此所需的配置信息也不同,需要通过neutron组件的原来的配置文件进行分析,给neutron组件的各个节点注册自己关联的配置信息,下面通过一个具体的实施例对配置信息注册过程进行介绍。

[0139] 如图5所示,分别将各个节点neutron-server、neutron-dhcp-agent、neutron-openvswitch-agent的配置项和相对应的配置信息修改函数进行注册,图中的配置文件分别表示各个节点对应的配置项以及配置信息修改函数(如图中所示的neutron-server相关的配置、neutron-dhcp-agent相关的配置、neutron-openvswitch-agent相关的配置),各个节点的配置更新子系统均可以包括配置注册模块(图中所示的配置注册)、配置key匹配模块(图中所示的配置key匹配)、以及配置处理模块(图中所示的配置处理),各个节点对应的配置注册模块分别用于将neutron-server相关的配置、neutron-dhcp-agent相关的配置、neutron-openvswitch-agent相关的配置注册到相应的配置注册模块。当配置更新子系统接收到配置更新请求时,针对于每个节点,其中的配置key匹配模块用于将配置更新请求中解析出的配置项的key的值和配置注册模块中已注册的配置项的key的值进行匹配,匹配成功的配置项,以及配置项对应的配置信息修改函数发送至配置处理模块;配置处理模块用于利用配置项对应的配置信息修改函数将配置项的当前配置信息修改为新的配置信息。

[0140] 步骤S104,通过回调目标配置项各自对应的配置信息修改函数,将各目标配置项的当前配置信息更新为新的配置信息。

[0141] 具体的,可以通过回调的方式调用待更新的配置项对应的配置信息修改函数,利用修改函数将待更新的配置项的当前配置信息更新为新的配置信息。

[0142] 目标配置项的新的配置信息不同,利用配置信息修改函数对目标配置项进行更新的实现方式也不同,具体见如下实施例。

[0143] 在一种可能的实现方式中,对于一目标配置项,配置信息修改函数包括以下任一项:

[0144] 键值修改函数;

[0145] 键值修改函数和配置项处理函数;

[0146] 通过回调该目标配置项对应的配置信息修改函数,将该目标配置项的当前配置信息更新为新的配置信息,包括:

[0147] 若配置信息修改函数为键值修改函数,则通过回调该目标配置项对应的键值修改函数,将目标配置项的单例对象的键值对中的值由当前配置信息修改为新的配置信息;

[0148] 若所述配置信息修改函数为键值修改函数和配置项处理函数,则通过回调该目标配置项对应的键值修改函数,将目标配置项的单例对象的键值对中的值由当前配置信息修改为新的配置信息,并通过回调该目标配置项对应的配置项处理函数,将该目标配置项当前配置信息更新为新的配置信息。

[0149] 在实际应用中,配置信息修改函数可以包括键值修改函数,或者键值修改函数和配置项处理函数,对配置项进行注册时,为每个配置项配置了对应的配置信息修改函数,有些配置项的配置信息修改函数为键值修改函数,有些配置项的配置信息修改函数为键值修改函数和配置项处理函数。其中,配置项处理函数是对配置项进行相应处理的函数,例如,线程创建函数等。如果该目标配置项的配置信息更新是针对目标配置项的键值对中的值进

行更新的,也就是说,修改目标配置项对应的键值对中的值,即可实现该目标配置项的配置更新,则调用目标配置项对应的键值修改函数,修改存储目标配置项的单例对象的键值对中的值,将键值对中的值由当前配置信息修改为新的配置信息。

[0150] 如果目标配置项的更新既需要修改键值对,又需要通过回调相应的函数进行修改,例如,修改配置项对应的工作线程数量,则通过回调的方式调用目标配置项对应的键值修改函数,以及目标配置项的配置项处理函数,利用键值修改函数修改该目标配置项,并且利用配置项处理函数进行相应处理,例如,通过线程创建函数创建新的工作线程、利用线程关闭函数关闭某些工作线程。除了工作线程的修改之外,还可以包括通过回调其他函数来对目标配置项进行更新,本申请对此不做限定。

[0151] 在一示例中,目标配置项的当前配置信息为“pool_size=5”,新的配置信息为“pool_size=10”,则通过回调的方式调用键值修改函数,将原来得“pool_size=5”修改为“pool_size=10”,然后调用创建新线程的线程创建函数,创建5个新的工作线程。

[0152] 在又一示例中,目标配置项的当前配置信息为“pool_size=5”,新的配置信息为“pool_size=3”,则通过回调键值修改函数将“pool_size=5”改为“pool_size=3”;对于当前即将完成工作的两个线程,不再分配新的工作,通过回调的方式调用线程关闭函数,具体可以是一条关闭线程的指令,待这两个工作线程工作完成时,关闭这两个工作线程。

[0153] 下面通过一个具体的实施例,对配置更新子系统中各个模块的具体工作过程进行描述。

[0154] 如图6所示,配置更新子系统配置解析模块(图中所示的配置解析)解析从restful api或者套接字监测模块中得到配置更新请求,得到待更新的各目标配置项以及各目标配置项的新的配置信息,发送到配置匹配模块(图中所示的配置匹配);配置注册模块(图中所示的配置注册)中预先注册了需要动态更新的各配置项以及各配置项对应的配置信息修改函数;配置匹配模块将各目标配置项和配置注册模块中注册的配置项进行匹配,将匹配成功的各配置项的配置信息修改函数作为相应的各目标配置项对应的配置信息修改函数;内存应用模块(图中所示的内存应用)通过回调的方式调用各目标配置项对应的配置信息修改函数,利用配置信息修改函数将目标配置项的当前配置信息进行修改,将待更新的各目标配置项的新的配置信息应用到内存中。其中,每个回调对应的配置信息修改函数不同,以3个回调为例(如图中所示的回调1、回调2、回调3),每个回调调用一个配置信息修改函数,利用各配置信息修改函数将目标配置项中当前配置信息修改为新的配置信息。

[0155] 本申请实施例提供的配置信息更新方法,首先获取配置更新请求;解析配置更新请求,得到更新配置信息,更新配置信息包括待更新的各目标配置项的标识以及各目标配置项的新的配置信息;确定与各目标配置项各自对应的配置信息修改函数;通过回调目标配置项各自对应的配置信息修改函数,将各目标配置项的当前配置信息更新为新的配置信息。本申请提供的技术方案,在对neutron服务组件进行配置更新时,通过回调目标配置项各自对应的配置信息修改函数,来实现将当前配置信息更新为新的配置信息,不需要重启neutron服务组件,避免了由于重启造成的虚拟机网络瘫痪问题,不影响neutron服务组件的正常工作,提升了系统的稳定性。

[0156] 下面以控制节点根据接收到的Restful请求进行配置信息更新的具体实现过程进

行介绍。

[0157] 本申请实施例提供了一种配置信息更新方法,如图7所示,该方法包括:

[0158] 步骤S201,接收Restful请求,解析Restful请求,得到各节点标识和每个节点标识所对应的配置更新请求,其中,配置更新请求中包括更新配置信息,更新配置信息包括待更新的各目标配置项、以及各目标配置项的新的配置信息;

[0159] 具体的,控制节点通过server进程接收restful请求,restful请求中携带有各节点标识以及每个节点标识对应的配置更新请求,对restful请求进行解析之后,得到各节点标识和每个节点标识所对应的配置更新请求,配置更新请求可以是针对neutron服务组件中的控制节点或者计算节点中的配置信息进行更新的请求。配置更新请求中携带有待更新的各目标配置项、以及各目标配置项的新的配置信息。

[0160] 步骤S202,若节点标识包括控制节点的标识,则确定与控制节点的标识对应的各目标配置项各自对应的配置信息修改函数,并通过回调目标配置项各自对应的配置信息修改函数,将各目标配置项的当前配置信息更新为新的配置信息;

[0161] 如果节点标识包括控制节点的标识,则表示配置更新请求是针对控制节点的,则确定与控制节点的标识对应的各目标配置项各自对应的配置信息修改函数,通过回调目标配置项各自对应的配置信息修改函数,将各目标配置项的当前配置信息更新为新的配置信息。

[0162] 步骤S203,若节点标识包括neutron代理节点的标识,则将配置更新请求发送至对应的neutron代理节点,以使neutron代理节点根据接收到的配置更新请求,确定该配置更新请求中的各目标配置项各自对应的配置信息修改函数,并通过回调目标配置项各自对应的配置信息修改函数,将各目标配置项的当前配置信息更新为新的配置信息。

[0163] 如果节点标识包括neutron代理节点的标识,则表示配置更新请求是针对代理节点的,其中,代理节点可以包括控制节点的代理节点或者计算节点的代理节点,控制节点根据代理节点的标识将配置更新请求发送至对应的代理节点,代理节点确定该配置更新请求中的各目标配置项各自对应的配置信息修改函数,并通过回调目标配置项各自对应的配置信息修改函数,将各目标配置项的当前配置信息更新为新的配置信息。

[0164] 步骤S201-步骤S203的具体实现过程已经在图1对应的实施例中的多个可选实施例中进行了详细描述,此处不再赘述。

[0165] 本申请提供的技术方案,在对neutron服务组件进行配置更新时,通过回调目标配置项各自对应的配置信息修改函数,来实现将当前配置信息更新为新的配置信息,不需要重启neutron服务组件,避免了由于重启造成的虚拟机网络瘫痪问题,不影响neutron服务组件的正常工作,提升了系统的稳定性。

[0166] 下面通过一个具体的实施例,以流程图的方式对配置更新的具体过程进行描述。

[0167] 如图8所示,对于接收到的请求,如果是restful请求,则执行步骤S301,对restful请求进行解析,解析结果中包括各节点标识,以及各节点标识对应的配置更新请求,根据节点标识判断配置更新请求是否是针对neutron组件的控制节点的server进程的neutron-server配置更新请求,如果不是,则执行步骤S302,通过Rpc远程调用的方式将配置更新请求发送至节点标识对应的代理节点;执行步骤S303,代理节点的配置更新子系统进行处理;

[0168] 如果是命令行请求,则将该请求写入套接字文件,通过对套接字文件进行监测,得

到配置更新请求,执行步骤S303,由配置更新请求对应的节点的配置更新子系统进行处理;

[0169] 如果是neurton-server配置更新请求,则直接执行步骤S303,将该配置更新请求发送到控制节点的配置更新子系统进行处理;

[0170] 接下来步骤S304,配置更新子系统解析配置更新请求,得到各目标配置项和各目标配置项对应的新的配置信息;

[0171] 步骤S305,将各目标配置项和配置注册模块中注册的配置项进行匹配,将匹配成功的各配置项的配置信息修改函数作为相应的各目标配置项对应的配置信息修改函数;

[0172] 步骤S306,通过回调的方式回调目标配置项的键值修改函数,将各目标配置项的存储在内存中的键值对进行修改,将新的配置信息中的键值对应用到内存中;

[0173] 步骤S307,通过回调的方式回调相应的配置项处理函数,将目标配置项的当前配置信息改为新的配置信息。

[0174] 需要说明的是,步骤S303中的请求处理子系统进行处理步骤,是配置更新请求对应的节点的配置更新子系统进行处理,可能是控制节点的配置更新子系统,也可能是代理节点的配置更新子系统,由于各节点中的处理更新子系统的结构和功能相同,因此图中未进行区分。

[0175] 下面通过一个具体的实施例,对本申请技术方案所适用的一种系统架构进行描述。

[0176] 如图9所示,系统架构分为三层,各层的具体介绍如下:

[0177] 第一层是业务层,包括自助平台、运维平台以及监测平台。

[0178] 自助平台用于向用户提供申请网络资源的界面,用户可以通过自助平台申请网络资源,网络资源可以包括:虚拟网卡、虚拟路由器、虚拟防火墙、虚拟交换机等。

[0179] 运维平台用于向用户提供管理员操作的界面,通过该界面可以进行宿主机的迁移等操作,将网络资源中的数据转存储到指定的设备中。

[0180] 监测平台用于监测整个平台的全部网络资源使用情况。

[0181] 第二层是网关层,包括:数据库(图中所示的DB)、控制模块(图中所示的venus)、网络协议(Internet Protocol,IP)管理系统(图中所示的IP系统)以及基于角色的访问控制Rbac模块(图中所示的Rbac);

[0182] 数据库用于存储网络资源相关的数据;

[0183] 控制模块可以通过venus操作系统来实现,用于向业务层中的自助平台、运维平台和监测平台提供统一的资源接口,并且与数据库、IP管理系统、Rbac模块进行通信;

[0184] IP管理系统用于管理网络协议等网络资源;

[0185] Rbac模块用于为不同的用户分配对应的操作权限。

[0186] 第三层是基础云层(图中所示的Openstack层),包括Rest Client、Cli Client以及neutron组件(图中所示的neutron);

[0187] 基础云层主要用于为开源的Openstack提供IaaS服务。

[0188] Rest Client用于通过restful api接收来自于网络层的配置更新请求,请求的类型可以包括POST请求、GET请求以及LIST请求;

[0189] Cli Client用于向用户提供命令行界面,接收用户输入的命令行形式的配置更新请求,可以对neutron组件中的配置项的键值对“key=value”进行设置(如图中所示的

neutron config set key=value);

[0190] neutron组件中包含配置更新子系统,通过配置更新子系统实现neutron组件中各个节点的配置信息更新。

[0191] 其中,基础云层的neutron组件通过两种方式使用配置更新子系统,一种是restful请求,一种是命令行请求。一般restful请求可以提供给私有云的运维系统,供运维管理人员使用;restful client还可以提供给任意的第三方调用,比如配置中心。命令行请求可以在后台服务提供给开发人员使用。通过在产品上提供两种使用方式,使配置更新使用起来更加的方便,适合满足更多的需求。

[0192] 基于与图1中所示方法相同的原理,本公开的实施例中还提供了一种配置信息更新装置40,如图10所示,该配置信息更新装置40包括:

[0193] 获取模块41,用于获取配置更新请求;

[0194] 解析模块42,用于解析配置更新请求,得到更新配置信息,更新配置信息包括待更新的各目标配置项的标识、以及各目标配置项的新的配置信息;

[0195] 确定模块43,用于确定与各目标配置项各自对应的配置信息修改函数;

[0196] 更新模块44,用于通过回调目标配置项各自对应的配置信息修改函数,将各目标配置项的当前配置信息更新为新的配置信息。

[0197] 在一种可能的实现方式中,该装置40被配置于neutron代理节点中,获取模块41可以包括以下至少一项:

[0198] 第一接收单元,用于接收neutron控制节点发送的配置更新请求,其中,配置更新请求为neutron控制节点通过接收到的第一表现层状态转化restful请求确定出的;

[0199] 第一监测单元,用于通过监测第一Unix套接字文件获取配置更新请求。

[0200] 在一种可能的实现方式中,装置40还包括第一接收模块;

[0201] 第一接收模块,用于接收第一命令行CLI配置更新请求,将接收到的配置更新请求写入第一Unix套接字文件。

[0202] 在一种可能的实现方式中,neutron代理节点包括neutron控制节点的代理节点或neutron计算节点的代理节点。

[0203] 在一种可能的实现方式中,第一restful请求中携带有neutron代理节点的节点标识,配置更新请求为neutron控制节点通过以下方式确定出的:

[0204] 接收第一Restful请求;

[0205] 解析第一Restful请求,基于解析结果得到各节点标识和每个节点标识所对应的配置更新请求。

[0206] 在一种可能的实现方式中,装置40被配置于neutron控制节点中,获取模块41可以包括以下至少一项:

[0207] 第二接收单元,用于接收第二Restful请求,基于第二Restful请求得到配置更新请求;

[0208] 第二监测单元,用于通过监测第二Unix套接字文件获取配置更新请求。

[0209] 在一种可能的实现方式中,装置40还包括第二接收模块;

[0210] 第二接收模块,用于接收第二CLI配置更新请求,将接收到的配置更新请求写入第二Unix套接字文件。

[0211] 在一种可能的实现方式中,第二Restful请求中携带有neutron控制节点的节点标识,装置40还包括请求确定模块;

[0212] 请求确定模块,用于解析第二Restful请求,基于解析结果得到各节点标识和每个节点标识所对应的配置更新请求;

[0213] 将各节点标识中neutron控制节点的节点标识所对应的配置更新请求,确定为配置更新请求。

[0214] 在一种可能的实现方式中,装置40还包括配置项设置模块;

[0215] 配置项设置模块,用于接收配置项设置请求,配置项设置请求包括各可更新配置项的标识、以及各可更新配置项各自所对应的配置信息修改函数;

[0216] 根据配置项设置请求,建立各可更新配置项的标识、以及各可更新配置项各自所对应的配置信息修改函数之间的映射关系并存储;

[0217] 确定模块43,用于基于映射关系,以及各目标配置项的标识,确定与各目标配置项各自对应的配置信息修改函数。

[0218] 在一种可能的实现方式中,装置40还包括配置信息获取模块;

[0219] 配置信息获取模块,用于根据配置项设置请求中的各可更新配置项的标识,获取对应的各可更新配置项的当前配置信息;

[0220] 采用键值对形式的单例对象保存各可更新配置项的标识及对应的当前配置信息,其中,键值对中的键为可更新配置项的标识,值为可更新配置项的当前配置信息。

[0221] 在一种可能的实现方式中,对于一目标配置项,配置信息修改函数包括以下任一项:

[0222] 键值修改函数;

[0223] 键值修改函数和配置项处理函数;

[0224] 更新模块44,用于:

[0225] 若配置信息修改函数为键值修改函数,则通过回调该目标配置项对应的键值修改函数,将目标配置项的单例对象的键值对中的值由当前配置信息修改为新的配置信息;

[0226] 若所述配置信息修改函数为键值修改函数和配置项处理函数,通过回调该目标配置项对应的键值修改函数,将目标配置项的单例对象的键值对中的值由当前配置信息修改为新的配置信息,并通过回调该目标配置项对应的配置项处理函数,将该目标配置项的当前配置信息更新为新的配置信息。

[0227] 本申请实施例的装置与图1的方法相对应,具体的执行过程已经在图1对应的实施例进行了详细描述,此处不再赘述。

[0228] 本申请实施例提供的配置信息更新装置,获取配置更新请求;解析配置更新请求,得到更新配置信息,更新配置信息包括待更新的各目标配置项的标识以及各目标配置项的新的配置信息;确定与各目标配置项各自对应的配置信息修改函数;通过回调目标配置项各自对应的配置信息修改函数,将各目标配置项的当前配置信息更新为新的配置信息。本申请提供的技术方案,在对neutron服务组件进行配置更新时,通过回调目标配置项各自对应的配置信息修改函数,来实现将当前配置信息更新为新的配置信息,不需要重启neutron服务组件,避免了由于重启造成的虚拟机网络瘫痪问题,不影响neutron服务组件的正常工作,提升了系统的稳定性。

[0229] 基于与图7中所示方法相同的原理,本公开的实施例中还提供了一种配置信息更新装置50,如图11所示,该配置信息更新装置50包括:

[0230] 接收模块51,用于接收Restful请求,解析Restful请求,得到各节点标识和每个节点标识所对应的配置更新请求,其中,配置更新请求中包括更新配置信息,更新配置信息包括待更新的各目标配置项的标识、以及各目标配置项的新的配置信息;

[0231] 确定模块52,用于在节点标识包括neutron控制节点的标识时,则确定与neutron控制节点的标识对应的各目标配置项各自对应的配置信息修改函数,并通过回调目标配置项各自对应的配置信息修改函数,将各目标配置项的当前配置信息更新为新的配置信息;

[0232] 发送模块53,用于在节点标识包括neutron代理节点的标识时,将配置更新请求发送至对应的neutron代理节点,以使neutron代理节点根据接收到的配置更新请求,确定该配置更新请求中的各目标配置项各自对应的配置信息修改函数,并通过回调目标配置项各自对应的配置信息修改函数,将各目标配置项的当前配置信息更新为新的配置信息。

[0233] 本申请实施例的装置与图7的方法相对应,具体的执行过程已经在图7对应的实施例进行了详细描述,此处不再赘述。

[0234] 上述实施例从虚拟模块的角度介绍了配置信息更新装置,下述从实体模块的角度介绍一种电子设备,具体如下所示:

[0235] 本申请实施例提供了一种电子设备,如图12所示,图12所示的电子设备9000包括:处理器9001和存储器9003。其中,处理器9001和存储器9003相连,如通过总线9002相连。可选地,电子设备9000还可以包括收发器9004。需要说明的是,实际应用中收发器9004不限于一个,该电子设备9000的结构并不构成对本申请实施例的限定。

[0236] 处理器9001可以是CPU,通用处理器,GPU,DSP,ASIC,FPGA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9001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。

[0237] 总线9002可包括一通路,在上述组件之间传送信息。总线9002可以是PCI总线或EISA总线等。总线9002可以分为地址总线、数据总线、控制总线等。为便于表示,图12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。

[0238] 存储器9003可以是ROM或可存储静态信息和其他类型的静态存储设备,RAM或者可存储信息和其他类型的动态存储设备,也可以是EEPROM、CD-ROM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)、磁盘存储介质或者其他磁存储设备、或者能够用于携带或存储具有指令或数据结构形式的期望的程序代码并能够由计算机存取的任何其他介质,但不限于此。

[0239] 存储器9003用于存储执行本申请方案的应用程序代码,并由处理器9001来控制执行。处理器9001用于执行存储器9003中存储的应用程序代码,以实现前述任一方法实施例所示的内容。

[0240] 本申请实施例提供了一种电子设备,本申请实施例中的电子设备包括:一个或多个处理器;存储器;一个或多个计算机程序,其中一个或多个计算机程序被存储在存储器中并被配置为由一个或多个处理器执行,一个或多个程序被处理器执行时,获取配置更新请求;解析配置更新请求,得到更新配置信息,更新配置信息包括待更新的各目标配置项、以

及各目标配置项的新的配置信息;确定与各目标配置项各自对应的配置信息修改函数;通过回调目标配置项各自对应的配置信息修改函数,将各目标配置项的当前配置信息更新为新的配置信息。或者

[0241] 接收Restful请求,解析Restful请求,得到各节点标识和每个节点标识所对应的配置更新请求,其中,配置更新请求中包括更新配置信息,更新配置信息包括待更新的各目标配置项、以及各目标配置项的新的配置信息;若节点标识包括neutron控制节点的标识,则确定与neutron控制节点的标识对应的各目标配置项各自对应的配置信息修改函数,并通过回调目标配置项各自对应的配置信息修改函数,将各目标配置项的当前配置信息更新为新的配置信息;若节点标识包括neutron代理节点的标识,则将配置更新请求发送至对应的neutron代理节点,以使neutron代理节点根据接收到的配置更新请求,确定该配置更新请求中的各目标配置项各自对应的配置信息修改函数,并通过回调目标配置项各自对应的配置信息修改函数,将各目标配置项的当前配置信息更新为新的配置信息。

[0242] 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,当计算机程序在处理器上运行时,使得处理器可以执行前述方法实施例中相应内容。

[0243] 应该理解的是,虽然附图的流程图中的各个步骤按照箭头的指示依次显示,但是这些步骤并不是必然按照箭头指示的顺序依次执行。除非本文中有明确的说明,这些步骤的执行并没有严格的顺序限制,其可以以其他的顺序执行。而且,附图的流程图中的至少一部分步骤可以包括多个子步骤或者多个阶段,这些子步骤或者阶段并不必然是在同一时刻执行完成,而是可以在不同的时刻执行,其执行顺序也不必然是依次进行,而是可以与其他步骤或者其他步骤的子步骤或者阶段的至少一部分轮流或者交替地执行。

[0244] 以上所述仅是本发明的部分实施方式,应当指出,对于本技术领域的普通技术人员来说,在不脱离本发明原理的前提下,还可以做出若干改进和润饰,这些改进和润饰也应视为本发明的保护范围。

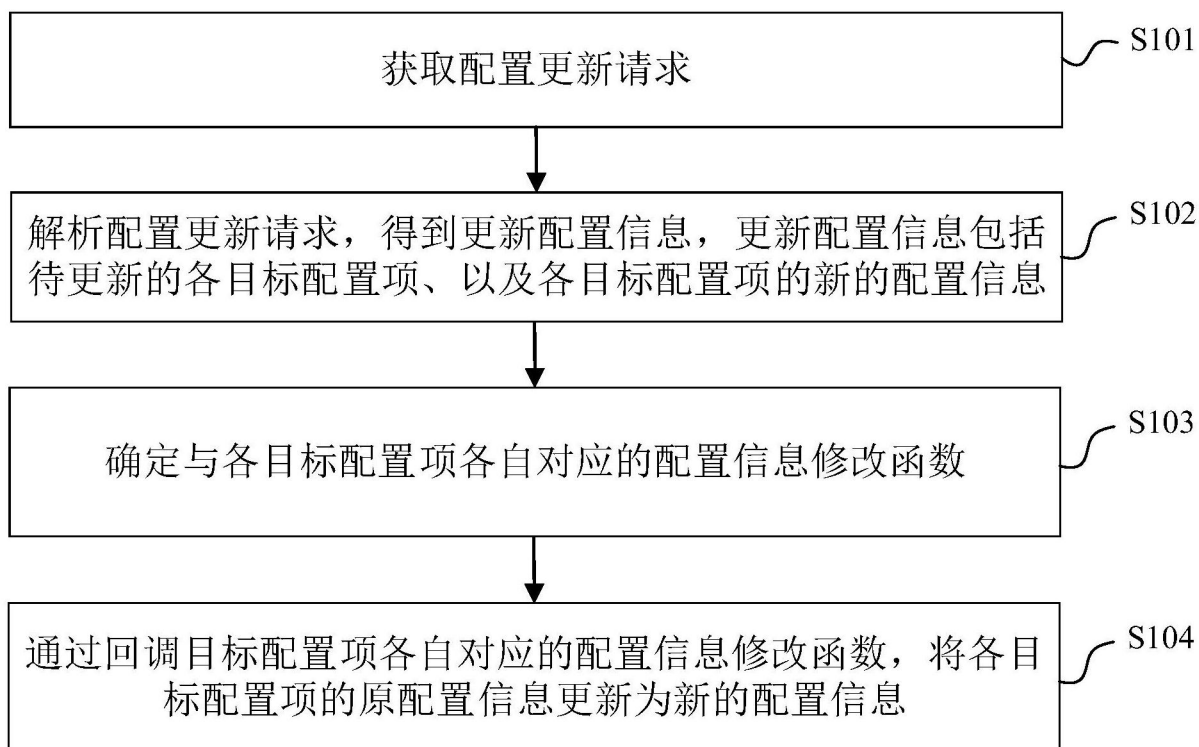


图1

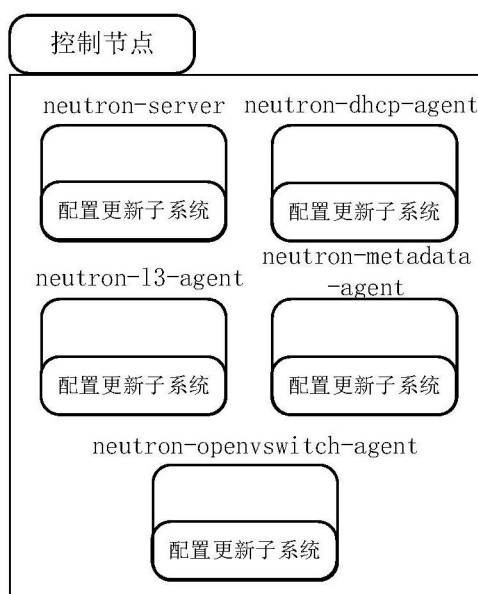


图2a



图2b

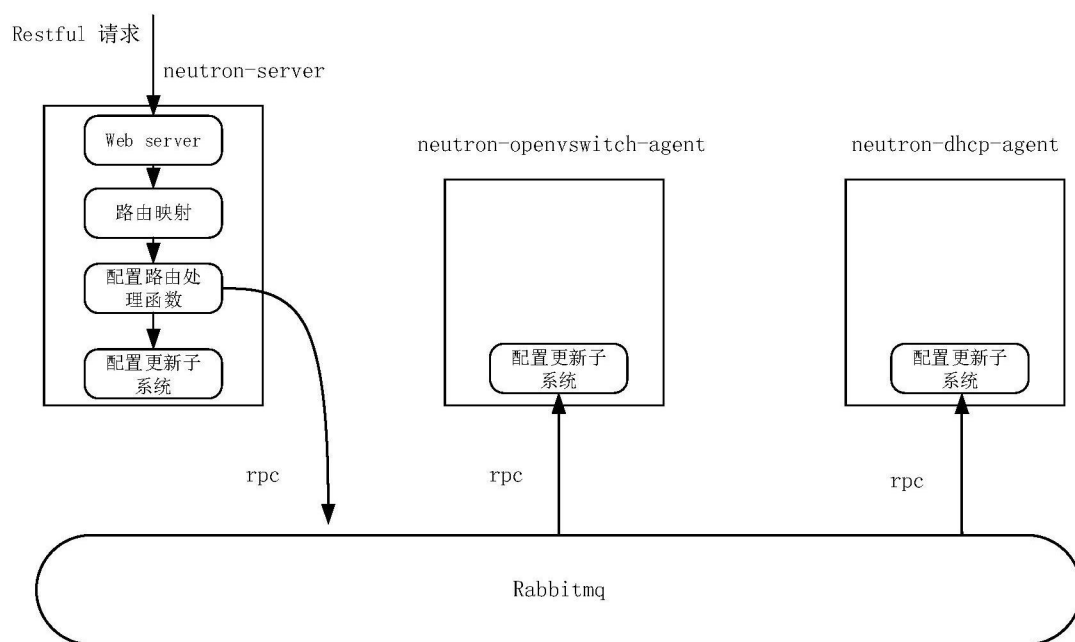


图3

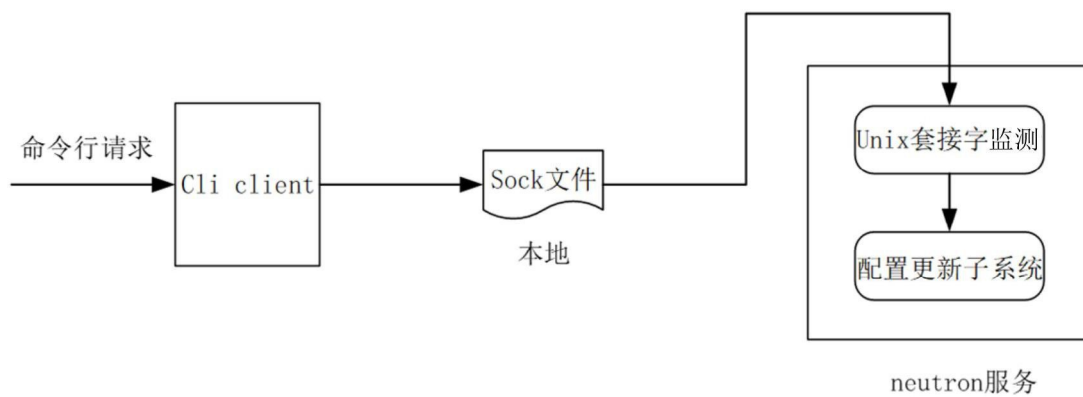


图4

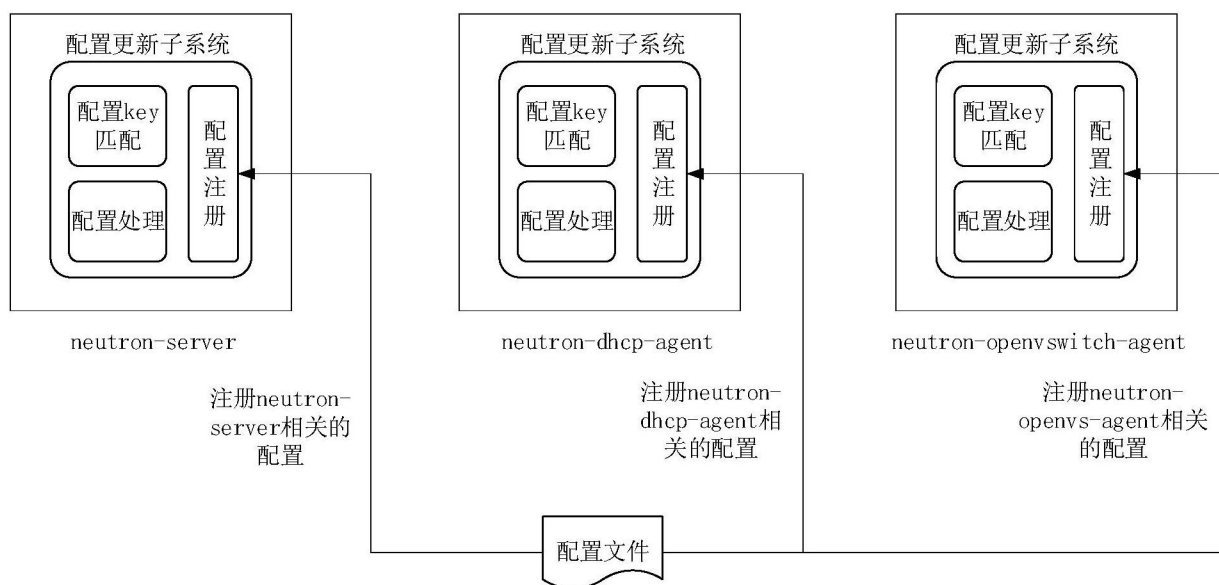


图5

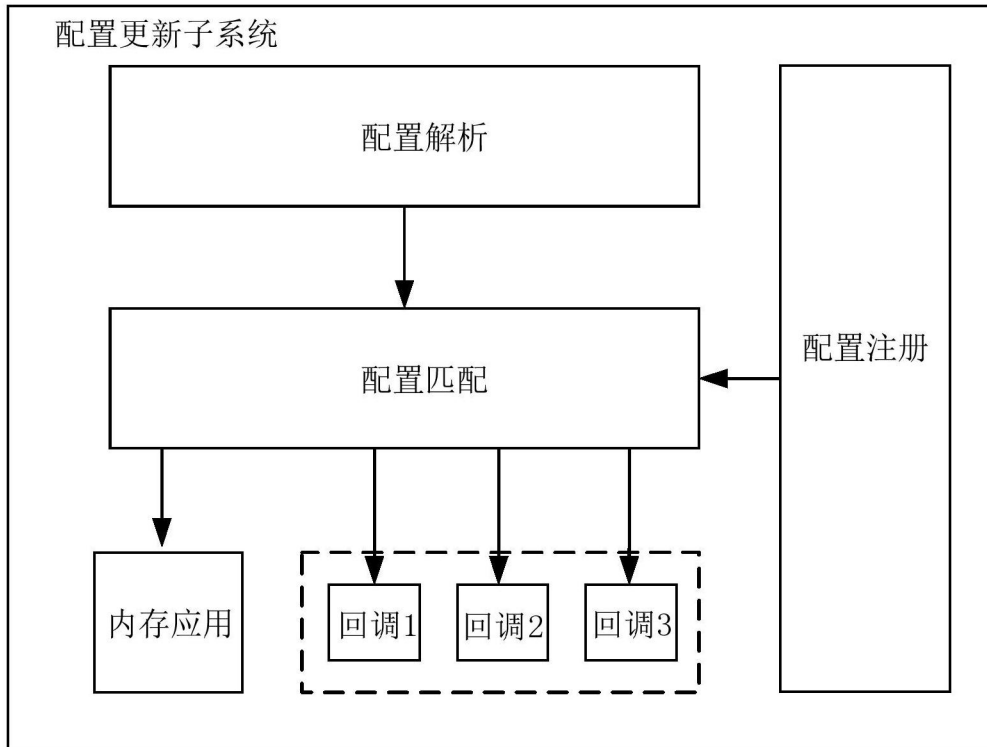


图6

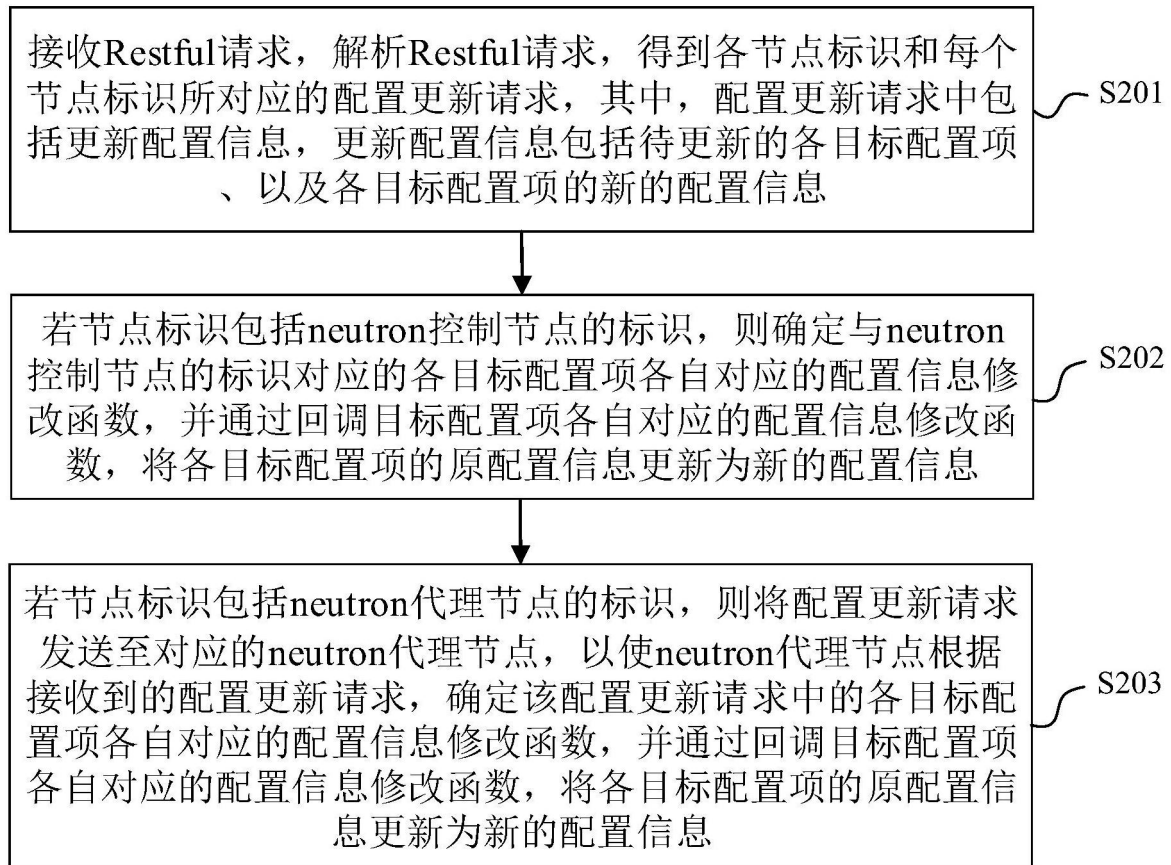


图7

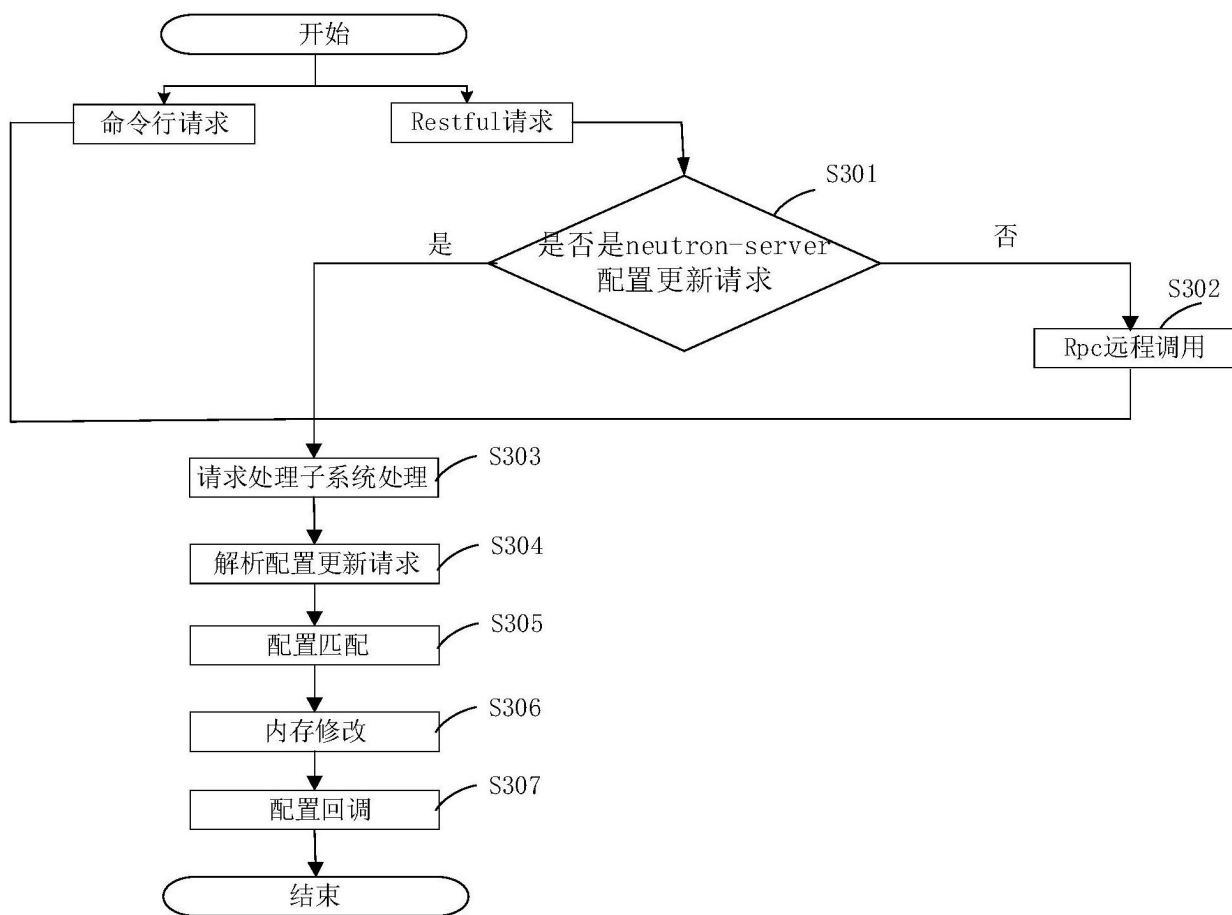


图8

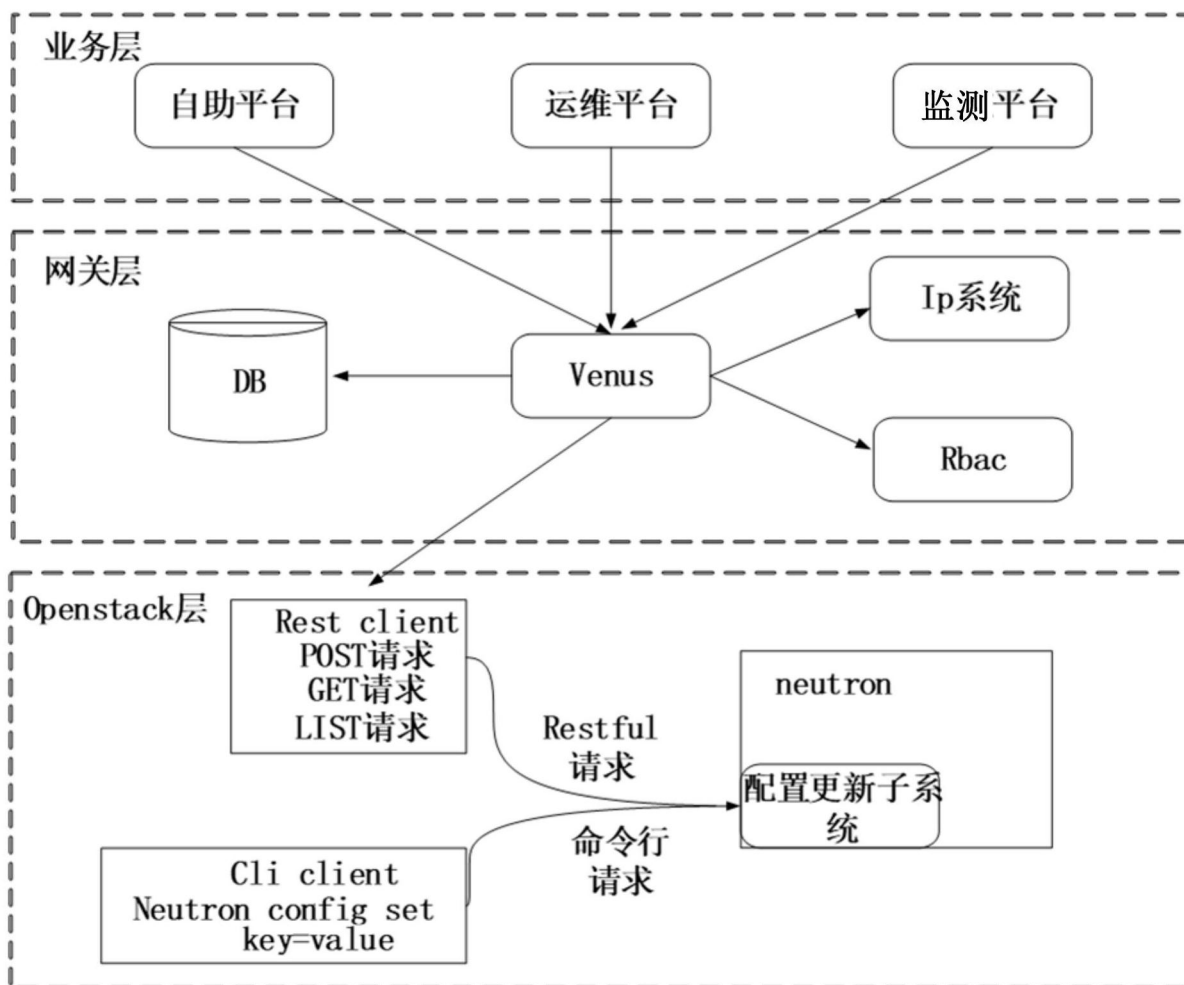


图9

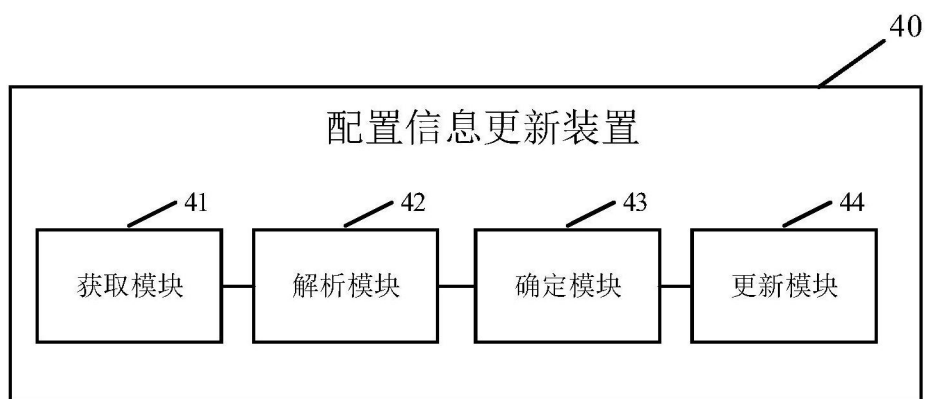


图10

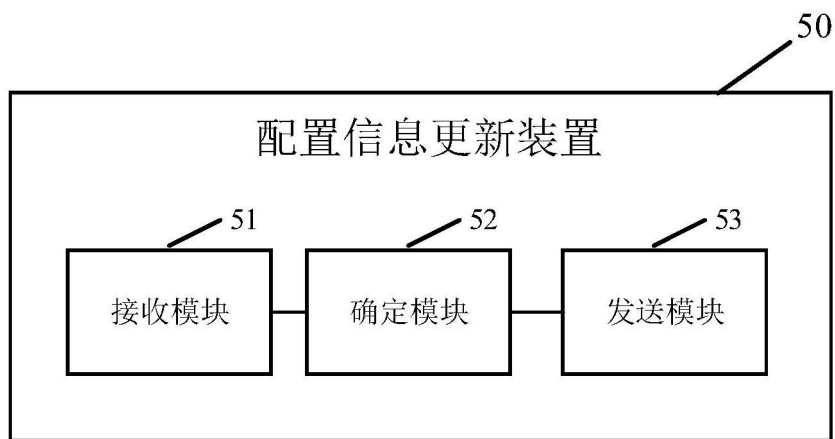


图11

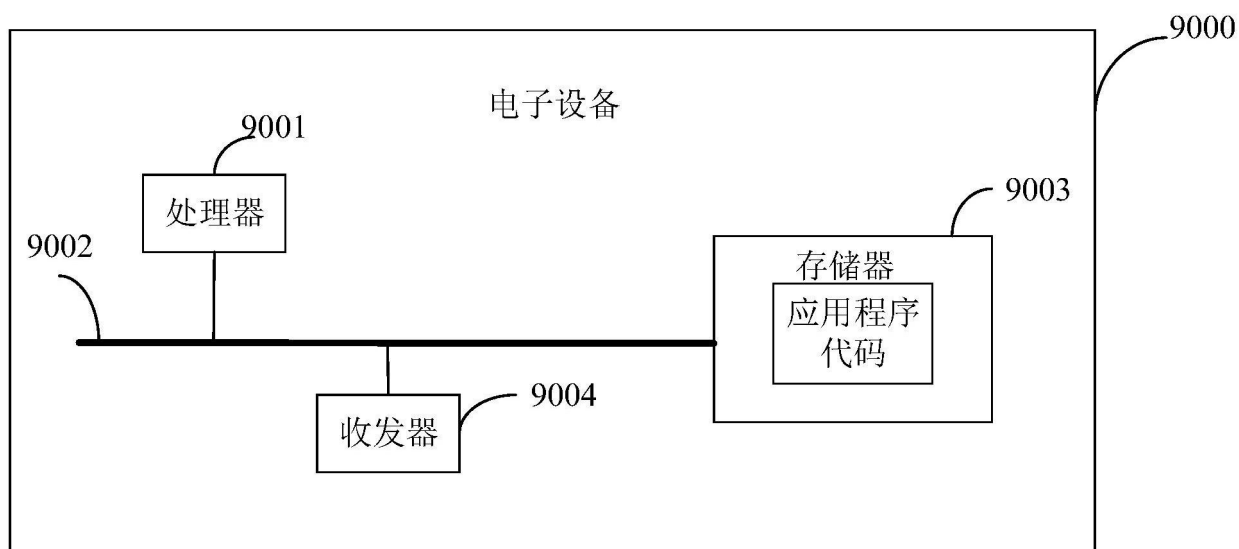


图12