



(12) 发明专利申请

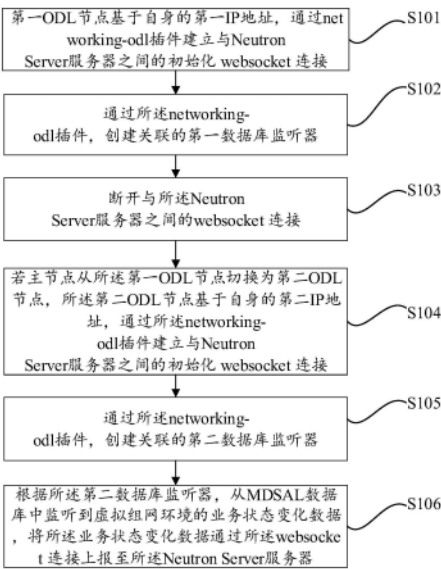
(10) 申请公布号 CN 116915834 A
(43) 申请公布日 2023. 10. 20

(21) 申请号 202310889989.5
(22) 申请日 2023.07.19
(71) 申请人 浪潮思科网络科技有限公司
地址 250101 山东省济南市高新区浪潮路
1036号S01-6楼
(72) 发明人 马章竞
(74) 专利代理机构 北京君慧知识产权代理事务
所(普通合伙) 11716
专利代理师 肖鹏
(51) Int.Cl.
H04L 67/141 (2022.01)
H04L 67/143 (2022.01)
H04L 67/51 (2022.01)
G06F 16/27 (2019.01)

权利要求书3页 说明书8页 附图3页

(54) 发明名称
一种ODL控制器的业务状态上报方法、设备
及介质
(57) 摘要

本申请公开了一种ODL控制器的业务状态上报方法、设备及介质,方法包括:第一ODL节点基于自身的第一IP地址,通过networking-odl插件建立与Neutron Server服务器之间的初始化 websocket连接;其中,第一ODL节点为主节点;通过networking-odl插件,创建关联的第一数据库监听器;断开与Neutron Server服务器之间的websocket连接;主节点从第一ODL节点切换为第二ODL节点,第二ODL节点基于自身的第二IP地址,通过networking-odl插件建立与Neutron Server服务器之间的websocket连接;通过networking-odl插件,创建关联的第二数据库监听器;根据第二数据库监听器,从MDSAL数据库中监听到虚拟组网环境的业务状态变化数据,将业务状态变化数据通过websocket连接上报至Neutron Server服务器。能够在切换主节点后,自动将虚拟业务状态数据上报。



CN 116915834 A

1. 一种ODL控制器的业务状态上报方法,其特征在于,所述方法包括:

第一ODL节点基于自身的第一IP地址,通过networking-odl插件建立与Neutron Server服务器之间的初始化websocket连接;其中,所述第一ODL节点为主节点;

通过所述networking-odl插件,创建关联的第一数据库监听器;

断开与所述Neutron Server服务器之间的websocket连接;

若主节点从所述第一ODL节点切换为第二ODL节点,所述第二ODL节点基于自身的第二IP地址,通过所述networking-odl插件建立与Neutron Server服务器之间的初始化websocket连接;

通过所述networking-odl插件,创建关联的第二数据库监听器;

根据所述第二数据库监听器,从MDSAL数据库中监听到虚拟组网环境的业务状态变化数据,将所述业务状态变化数据通过所述websocket连接上报至所述Neutron Server服务器。

2. 根据权利要求1所述的方法,其特征在于,所述根据所述第二数据库监听器,从MDSAL数据库中监听到虚拟组网环境的业务状态变化数据,具体包括:

在预设周期内,所述第二ODL节点通过MDSAL数据库的监听机制,提取物理网络设备的预设监听类型的配置信息;

根据所述配置信息,确定在虚拟组网环境中对应的虚拟业务状态数据,将所述虚拟业务状态数据存储至MDSAL数据库;

在所述MDSAL数据库中,将所述虚拟业务状态数据与历史虚拟业务状态数据进行对比;

若不一致,生成所述虚拟组网环境的业务状态变化数据,根据所述第二数据库监听器,监听到虚拟组网环境的业务状态变化数据。

3. 根据权利要求2所述的方法,其特征在于,所述根据所述配置信息,确定在虚拟组网环境中对应的虚拟业务状态数据,将所述虚拟业务状态数据存储至MDSAL数据库之前,所述方法还包括:

所述第二ODL节点,为所述MDSAL数据库创建兼容层;

通过所述兼容层,设置针对所述物理网络设备的监听字段类型,以注册针对所述MDSAL数据库中的字段监听。

4. 根据权利要求2所述的方法,其特征在于,所述配置信息为文本信息,所述第二ODL节点通过MDSAL数据库的监听机制,提取物理网络设备的预设监听类型的配置信息,具体包括:

所述第二ODL节点对所述配置信息进行解析,得到以树状结构为显示形式的树状配置信息;

通过正则表达式,在所述树状配置信息中,提取物理网络设备的预设监听类型的配置信息。

5. 根据权利要求4所述的方法,其特征在于,所述第二ODL节点对所述配置信息进行解析,得到以树状结构为显示形式的树状配置信息,具体包括:

所述第二ODL节点对所述配置信息进行遍历时,若检测到父节点,则为所述父节点生成树状结构组;

判断所述父节点是否包括子节点;

若是,则将所述子节点划分至所述树状结构组;

在遍历完成后,生成多个树状结构组,得到以树状结构为显示形式的树状配置信息。

6.根据权利要求1所述的方法,其特征在于,所述断开与所述Neutron Server服务器之间的websocket连接,具体包括:

若所述第一ODL节点处于非故障状态掉线时,或处于故障状态无法运行时,则断开与所述Neutron Server服务器之间的websocket连接;

所述通过所述networking-odl插件,创建关联的第二数据库监听器之后,所述方法还包括:

在所述第一ODL节点从掉线状态转换为在线状态时,若检测到所述第一数据库监听器出现报错信息,则将所述第一数据库监听器进行删除。

7.根据权利要求1所述的方法,其特征在于,所述将所述业务状态变化数据通过所述websocket连接上报至所述Neutron Server服务器之后,所述方法还包括:

在预先构建的映射关系表中,检索所述业务状态数据变化的字段类型对应的管理界面显示模型;

基于所述管理界面模型,将所述业务状态数据变化在管理界面进行状态显示。

8.根据权利要求2所述的方法,其特征在于,所述方法还包括:

若所述虚拟业务状态数据与所述历史虚拟业务状态数据一致,所述第二ODL节点则计算所述虚拟业务状态数据的获取时间与所述历史虚拟业务状态数据的获取时间之间的时间差;

对所述时间差与预设周期的时间进行求取差值;

若所述差值大于预设差值阈值,则确定所述第一ODL节点和第二ODL节点存在同时发生暂停工作的情况。

9.一种ODL控制器的业务状态上报设备,其特征在于,应用于第二ODL节点,包括:

至少一个处理器;以及,

与所述至少一个处理器通信连接的存储器;其中,

所述存储器存储有可被所述至少一个处理器执行的指令,所述指令被所述至少一个处理器执行,以使所述至少一个处理器能够:

若主节点从第一ODL节点切换为第二ODL节点,所述第二ODL节点基于自身的第二IP地址,通过所述networking-odl插件建立与Neutron Server服务器之间的初始化websocket连接;其中,所述第一ODL节点基于自身的第一IP地址,通过networking-odl插件建立与Neutron Server服务器之间的初始化websocket连接;通过所述networking-odl插件,创建关联的第一数据库监听器;断开与所述Neutron Server服务器之间的websocket连接;

通过所述networking-odl插件,创建关联的第二数据库监听器;

根据所述第二数据库监听器,从MDSAL数据库中监听到虚拟组网环境的业务状态变化数据,将所述业务状态变化数据通过所述websocket连接上报至所述Neutron Server服务器。

10.一种ODL控制器的业务状态上报非易失性计算机存储介质,存储有计算机可执行指令,其特征在于,应用于第二ODL节点,所述计算机可执行指令设置为:

若主节点从第一ODL节点切换为第二ODL节点,所述第二ODL节点基于自身的第二IP地

址,通过所述networking-odl插件建立与Neutron Server服务器之间的初始化websocket连接;其中,所述第一ODL节点基于自身的第一IP地址,通过networking-odl插件建立与Neutron Server服务器之间的初始化websocket连接;通过所述networking-odl插件,创建关联的第一数据库监听器;断开与所述Neutron Server服务器之间的websocket连接;

通过所述networking-odl插件,创建关联的第二数据库监听器;

根据所述第二数据库监听器,从MDSAL数据库中监听到虚拟组网环境的业务状态变化数据,将所述业务状态变化数据通过所述websocket连接上报至所述Neutron Server服务器。

一种ODL控制器的业务状态上报方法、设备及介质

技术领域

[0001] 本申请涉及SDN控制器技术领域,尤其涉及一种ODL控制器的业务状态上报方法、设备及介质。

背景技术

[0002] OpenDayLight (ODL) 是一个基于微服务架构的SDN控制器,支持广泛的网络协议,如OpenFlow、SNMP、NETCONF、OVSDB、BGP、LISP等。ODL具有良好的可编程性,采用OSGi Karaf作为底层实现服务模块化,并通过MDSAL和yang模型暴露模块接口。此模型内嵌了Level DB,包含了基于Akka Cluster的集群通信方案,功能丰富且开箱即用。

[0003] ODL的OpenStack ML2插件-networking-odl提供了路由端口状态更新的机制,简单来说,此插件会创建一个websocket服务端,ODL控制器启动后连接此websocket,并将neutron端口的MDSAL operational数据库数据变更事件发送到此websocket服务端,插件在接收到消息后,从中解析出端口状态:status,并且将其更新到Neutron Server中,实现Underlay路由端口状态上报。

[0004] 但是,基于硬件Overlay组网环境,在ODL集群环境下,由于networking-odl插件仅知道控制器集群的唯一虚拟IP,因此通过keepAlive等方案实现虚拟IP时,初始化websocket连接会在特定ODL节点创建数据库监听者,但是,在切换主节点后,websocket和新主节点重新建立连接后会找不到数据库监听器,无法自动将业务状态变化数据上报到Neutron Server服务器。

发明内容

[0005] 本申请实施例提供一种ODL控制器的业务状态上报方法、设备及介质,用于解决在切换主节点后,websocket和新主节点重新建立连接后会找不到数据库监听器,无法自动将业务状态变化数据上报到Neutron Server服务器的问题。

[0006] 本申请实施例采用下述技术方案:

[0007] 一方面,本申请实施例提供了一种ODL控制器的业务状态上报方法,该方法包括:第一ODL节点基于自身的第一IP地址,通过networking-odl插件建立与Neutron Server服务器之间的初始化websocket连接;其中,所述第一ODL节点为主节点;通过所述networking-odl插件,创建关联的第一数据库监听器;断开与所述Neutron Server服务器之间的websocket连接;若主节点从所述第一ODL节点切换为第二ODL节点,所述第二ODL节点基于自身的第二IP地址,通过所述networking-odl插件建立与Neutron Server服务器之间的初始化websocket连接;通过所述networking-odl插件,创建关联的第二数据库监听器;根据所述第二数据库监听器,从MDSAL数据库中监听到虚拟组网环境的业务状态变化数据,将所述业务状态变化数据通过所述websocket连接上报至所述Neutron Server服务器。

[0008] 一个示例中,所述根据所述第二数据库监听器,从MDSAL数据库中监听到虚拟组网环境的业务状态变化数据,具体包括:在预设周期内,所述第二ODL节点通过MDSAL数据库的

监听机制,提取物理网络设备的预设监听类型的配置信息;根据所述配置信息,确定在虚拟组网环境中对应的虚拟业务状态数据,将所述虚拟业务状态数据存储至MDSAL数据库;在所述MDSAL数据库中,将所述虚拟业务状态数据与所述历史虚拟业务状态数据进行对比;若不一致,生成所述虚拟组网环境的业务状态变化数据,根据所述第二数据库监听器,监听到虚拟组网环境的业务状态变化数据。

[0009] 一个示例中,所述根据所述配置信息,确定在虚拟组网环境中对应的虚拟业务状态数据,将所述虚拟业务状态数据存储至MDSAL数据库之前,所述方法还包括:所述第二ODL节点,为所述MDSAL数据库创建兼容层;通过所述兼容层,设置针对所述物理网络设备的监听字段类型,以注册针对所述MDSAL数据库中的字段监听。

[0010] 一个示例中,所述配置信息为文本信息,所述第二ODL节点通过MDSAL数据库的监听机制,提取物理网络设备的预设监听类型的配置信息,具体包括:所述第二ODL节点对所述配置信息进行解析,得到以树状结构为显示形式的树状配置信息;通过正则表达式,在所述树状配置信息中,提取物理网络设备的预设监听类型的配置信息。

[0011] 一个示例中,所述第二ODL节点对所述配置信息进行解析,得到以树状结构为显示形式的树状配置信息,具体包括:所述第二ODL节点对所述配置信息进行遍历时,若检测到父节点,则为所述父节点生成树状结构组;判断所述父节点是否包括子节点;若是,则将所述子节点划分至所述树状结构组;在遍历完成后,生成多个树状结构组,得到以树状结构为显示形式的树状配置信息。

[0012] 一个示例中,所述断开与所述Neutron Server服务器之间的websocket连接,具体包括:若所述第一ODL节点处于非故障状态掉线时,或处于故障状态无法运行时,则断开与所述Neutron Server服务器之间的websocket连接;所述通过所述networking-odl插件,创建关联的第二数据库监听器之后,所述方法还包括:在所述第一ODL节点从掉线状态转换为在线状态时,若检测到所述第一数据库监听器出现报错信息,则将所述第一数据库监听器进行删除。

[0013] 一个示例中,所述将所述业务状态变化数据通过所述websocket连接上报至所述Neutron Server服务器之后,所述方法还包括:在预先构建的映射关系表中,检索所述业务状态数据变化的字段类型对应的管理界面显示模型;基于所述管理界面模型,将所述业务状态数据变化在管理界面进行状态显示。

[0014] 一个示例中,所述方法还包括:若所述虚拟业务状态数据与所述历史虚拟业务状态数据一致,所述第二ODL节点则计算所述虚拟业务状态数据的获取时间与历史虚拟业务状态数据的获取时间之间的时间差;对所述时间差与预设周期的时间进行求取差值;若所述差值大于预设差值阈值,则确定所述第一ODL节点和第二ODL节点存在同时发生暂停工作的情况。

[0015] 另一方面,本申请实施例提供了一种ODL控制器的业务状态上报设备,应用于第二ODL节点,包括:至少一个处理器;以及,与所述至少一个处理器通信连接的存储器;其中,所述存储器存储有可被所述至少一个处理器执行的指令,所述指令被所述至少一个处理器执行,以使所述至少一个处理器能够:若主节点从第一ODL节点切换为第二ODL节点,所述第二ODL节点基于自身的第二IP地址,通过所述networking-odl插件建立与Neutron Server服务器之间的初始化websocket连接;其中,所述第一ODL节点基于自身的第一IP地址,通过

networking-odl插件建立与Neutron Server服务器之间的初始化websocket连接;通过所述networking-odl插件,创建关联的第一数据库监听器;断开与所述Neutron Server服务器之间的websocket连接;通过所述networking-odl插件,创建关联的第二数据库监听器;根据所述第二数据库监听器,从MDSAL数据库中监听到虚拟组网环境的业务状态变化数据,将所述业务状态变化数据通过所述websocket连接上报至所述Neutron Server服务器。

[0016] 另一方面,本申请实施例提供了一种ODL控制器的业务状态上报非易失性计算机存储介质,存储有计算机可执行指令,应用于第二ODL节点,所述计算机可执行指令设置为:若主节点从第一ODL节点切换为第二ODL节点,所述第二ODL节点基于自身的第二IP地址,通过所述networking-odl插件建立与Neutron Server服务器之间的初始化websocket连接;其中,所述第一ODL节点基于自身的第一IP地址,通过networking-odl插件建立与Neutron Server服务器之间的初始化websocket连接;通过所述networking-odl插件,创建关联的第一数据库监听器;断开与所述Neutron Server服务器之间的websocket连接;通过所述networking-odl插件,创建关联的第二数据库监听器;根据所述第二数据库监听器,从MDSAL数据库中监听到虚拟组网环境的业务状态变化数据,将所述业务状态变化数据通过所述websocket连接上报至所述Neutron Server服务器。

[0017] 本申请实施例采用的上述至少一个技术方案能够达到以下有益效果:

[0018] 通过networking-odl插件,将Neutron Server服务器与ODL节点的真实IP地址进行连接,从而在集群环境下,能够实现每次切换主节点时,WebSocket断开连接后,新主节点的ODL节点都能重新建立与Neutron Server服务器之间的初始化websocket连接,从而能够自动创建数据库监听器,实现能够和ODL节点关联特定的MDSAL数据库之间建立监听,继续获取Listener监听器对于MDSAL数据库变化的事件分派。

附图说明

[0019] 为了更清楚地说明本申请的技术方案,下面将结合附图来对本申请的部分实施例进行详细说明,附图中:

[0020] 图1为本申请实施例提供的一种ODL控制器的业务状态上报方法的流程示意图;

[0021] 图2为本申请实施例提供的一种应用场景下ODL控制器的业务状态上报系统的框架示意图;

[0022] 图3为本申请实施例提供的一种ODL控制器的业务状态上报设备的结构示意图。

具体实施方式

[0023] 为使本申请的目的、技术方案和优点更加清楚,下面将结合具体实施例及相应的附图对本申请的技术方案进行清楚、完整地描述。显然,所描述的实施例仅是本申请一部分实施例,而不是全部的实施例。基于本申请中的实施例,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例,都属于本申请保护的范围。

[0024] 下面参照附图来对本申请的一些实施例进行详细说明。

[0025] 图1为本申请实施例提供的一种ODL控制器的业务状态上报方法的流程示意图。该流程中的某些输入参数或者中间结果允许人工干预调节,以帮助提高准确性。

[0026] 图1中的流程可以包括以下步骤:

[0027] S101:第一ODL节点基于自身的第一IP地址,通过networking-odl插件建立与Neutron Server服务器之间的初始化websocket连接。

[0028] 其中,第一ODL节点为主节点。

[0029] 需要说明的是,networking-odl插件是云平台OpenStack调用SDN控制器的关键。neutron-server是提供neutron网络服务的服务器,任何有网络需求的个体发出的请求都会发给这样一个server进行网络请求。

[0030] S102:通过所述networking-odl插件,创建关联的第一数据库监听器。

[0031] S103:断开与所述Neutron Server服务器之间的websocket连接。

[0032] 其中,若第一ODL节点处于非故障状态掉线时,或处于故障状态无法运行时,则断开与Neutron Server服务器之间的websocket连接。比如,第一ODL节点关机时,则处于故障状态无法运行的状态。第一ODL节点由于网络中断,则处于非故障掉线的状态。

[0033] S104:若主节点从所述第一ODL节点切换为第二ODL节点,所述第二ODL节点基于自身的第二IP地址,通过所述networking-odl插件建立与Neutron Server服务器之间的初始化websocket连接。

[0034] S105:通过所述networking-odl插件,创建关联的第二数据库监听器。

[0035] 其中,在第一ODL节点从掉线状态转换为在线状态时,若检测到第一数据库监听器出现报错信息,则将第一数据库监听器进行删除。

[0036] S106:根据所述第二数据库监听器,从MDSAL数据库中监听到虚拟组网环境的业务状态变化数据,将所述业务状态变化数据通过所述websocket连接上报至所述Neutron Server服务器。

[0037] 在本申请的一些实施例中,在预先构建的映射关系表中,检索业务状态数据变化的字段类型对应的管理界面显示模型。基于管理界面模型,将业务状态数据变化在管理界面进行状态显示。

[0038] 在本申请的一些实施例中,在预设周期内,第二ODL节点通过MDSAL数据库的监听机制,提取物理网络设备的预设监听类型的配置信息。

[0039] 然后,根据配置信息,确定在虚拟组网环境中对应的虚拟业务状态数据,将虚拟业务状态数据存储至MDSAL数据库。

[0040] 在MDSAL数据库中,将虚拟业务状态数据与历史虚拟业务状态数据进行对比。

[0041] 若不一致,生成虚拟组网环境的业务状态变化数据,根据第二数据库监听器,监听到虚拟组网环境的业务状态变化数据。

[0042] 若一致,则不再生成虚拟组网环境的业务状态变化数据。

[0043] 此外,由于存在某种特殊需要,开发人员使得ODL节点集群需要暂停工作,此时,ODL节点集群并不能在预设周期内获取业务状态数据。

[0044] 基于此,在第二ODL节点在虚拟业务状态数据与历史虚拟业务状态数据一致时,第二ODL节点则计算虚拟业务状态数据的获取时间与历史虚拟业务状态数据的获取时间之间的时间差。

[0045] 对时间差与预设周期的时间进行求取差值;

[0046] 若差值大于预设差值阈值,则确定第一ODL节点和第二ODL节点存在同时发生暂停工作的情况,反之,则未出现存在同时发生暂停工作的情况。

[0047] 在本申请的一些实施例中,针对MDSAL数据库,借助了ODL内置数据库的监听机制,但在MDSAL数据库基础上提供了一个兼容层,不用为每个数据库都定义yang,然后为其手动创建集群监听并处理WebSocket消息,即,不需要为第二ODL节点创建多个不同yang的MDSAL数据库,而是借助于一个分片在不同节点上,即,多个ODL节点处于同一个分片,每个ODL节点关联数据库之间的内容相同,并且借助于raft算法底层同步的单独状态数据库,只需要简单注册对于此数据库某个字段的监听,即可使用全局的监听器实现其字段数据的变化事件生成。

[0048] 基于此,第二ODL节点,为MDSAL数据库创建兼容层;通过兼容层,设置针对所述物理网络设备的监听字段类型,以注册针对MDSAL数据库中的字段监听。

[0049] 在本申请的一些实施例中,第二ODL节点从物理网络设备提取的配置信息为文本信息,并不是直接采用正则表达式还原还原成树状结构,而是对文本信息进行树状命令建模,从而能够更加灵活准确地将物理网络设备预先设置的配置信息还原成树状结构,容错性更高。

[0050] 具体地,第二ODL节点对配置信息进行解析,得到以树状结构为显示形式的树状配置信息。

[0051] 其中,第二ODL节点对配置信息进行遍历时,若检测到父节点,则为父节点生成树状结构组;

[0052] 判断父节点是否包括子节点;若是,则将子节点划分至树状结构组。若否,则重新检测父节点。

[0053] 在遍历完成后,生成多个树状结构组,得到以树状结构为显示形式的树状配置信息。

[0054] 然后,通过正则表达式,在树状配置信息中,提取物理网络设备的预设监听类型的配置信息。

[0055] 本申请对于树状结构的配置进行快速、高效、兼容性解析,从而能够根据配置信息,得到虚拟服务状态,比如,路由端口的状态联通,必须依赖于其底层网络设备的路由配置正确:合适的vrf、vrf配置、vni和vlan映射等。对于非网络设备,比如防火墙硬件设备,大体也是基于树状结构返回的配置命令,因此这部分需要实现一种通用的树状结构配置解析器,而不能硬编码为正则表达式或字符串查找,后者很难正确处理。比如,part1 part1.1 part1.2 part1.3这类基于缩进的树状结构,如果判断part1下存在part1.1,直接正则查找即可,但有可能之后配置变更为part1 part1.0 part1.1 part1.2 part1.3,这将破坏正则规则。本发明实现的函数能够对树状命令建模:典型的比如show running-config,然后快速的查找某个父节点是否包含子节点,或某个子节点是否存在于某个父节点上,基于这种方式的状态检测鲁棒性更高,并且不用对每种硬件网络设备甚至是不同版本的网络设备进行单独适配,可靠性更高。

[0056] 提供了一层硬件Overlay组网环境下的树状命令状态检查、数据库监听和消息推送流程抽象,实现了,在数据采集端,相比较正则和字符串查找,树状命令解析的健壮性更好,并且不需要对不同版本的交换机命令行视频,多种型号的交换机或防火墙适配状态检查逻辑也很简单。数据库监听和消息推送流程相比较ODL提供的单节点,单数据库方案,更容易使用,没有模板代码,只需要两三行配置即可集成业务。并且对于ODL集群的集群主节

点切换适应良好。

[0057] 需要说明的是,虽然本申请实施例是参照图1来对步骤S101至步骤S106依次进行介绍说明的,但这并不代表步骤S101至步骤S106必须按照严格的先后顺序执行。本申请实施例之所以按照图1中所示的顺序对步骤S101至步骤S106依次进行介绍说明,是为了方便本领域技术人员理解本申请实施例的技术方案。换句话说,在本申请实施例中,步骤S101至步骤S106之间的先后顺序可以根据实际需要进行适当调整。

[0058] 通过图1的方法,通过networking-odl插件,将Neutron Server服务器与ODL节点的真实IP地址进行连接,从而在集群环境下,能够实现每次切换主节点时,WebSocket断开连接后,新主节点的ODL节点都能重新建立与Neutron Server服务器之间的初始化websocket连接,从而能够自动创建数据库监听器,实现能够和ODL节点关联特定的MDSAL数据库之间建立监听,继续获取Listener监听器对于MDSAL数据库变化的事件分派。

[0059] 图2为本申请实施例提供的一种应用场景下ODL控制器的业务状态上报系统的框架示意图。

[0060] 在图2中,通过提供一种在OpenDayLight+OpenStack平台上,基于Spine-Leaf硬件Overlay组网的SDN环境下,从底层网络设备的树状配置中提取虚拟组网环境的业务状态数据,之后借助于OpenDayLight的MDSAL数据库监听机制,通过更新状态到MDSAL数据库,从而触发Listener监听器和已建立WebSocket连接的Neutron Server交换数据,并在networking-odl插件中得到变化的状态,将其更新到OpenStack平台的抽象机制,这种机制可用于对任何有状态的虚拟服务,比如,将虚拟Overlay路由端口、VPN站点、VPN IPsec站点连接状态的业务状态数据快速更新到OpenStack或与OpenStack兼容的云平台,提升网络状态可观测性。

[0061] 通过图2的系统,实现OpenDayLight控制器集群中硬件Overlay组网环境中虚拟路由端口状态、VPN服务、VPN IPSEC站点连接等状态的上报功能。也就是说,基于基础的ODL状态通信机制实现一种抽象,此抽象可以对硬件Overlay环境中的交换机配置和状态进行针对性的各种业务检测,并可靠、稳定的上报到neutron server中,包括且不限于路由端口状态、VPN服务状态、VPN IPSEC树状命令解析的抽象;

[0062] 基于每个ODL节点单一数据库的数据更新监听与推送流程的抽象,WebSocket相比较REST请求更适合状态事件变更通知,开销更小。这种抽象可适配不同品牌、不同版本的交换机、负载均衡、防火墙等设备,通用性好,适配业务难度低,更健壮和稳定。站点连接状态等。

[0063] 比如,现有技术中,为每种型号每种系统版本的交换机和防火墙使用单独的代码获取其配置,然后正则和字符串解析,暴露为REST接口并让neutron server中的networking-odl进行轮询查找数据,对数据进行diff并获取状态变更信息。相比较本方案,这种方案缺乏抽象水平,各部分的耦合严重(交换机配置和代码、交换机版本和代码、解析状态变更的代码和发送状态变更的代码依赖数据库中转),且性能较差(neutron server集群和ODL集群轮询数据变更性能较差)。

[0064] 基于同样的思路,本申请的一些实施例还提供了上述方法对应的设备和非易失性计算机存储介质。

[0065] 图3为本申请实施例提供的一种ODL控制器的业务状态上报设备的结构示意图,应

用于第二ODL节点,包括:

[0066] 至少一个处理器;以及,

[0067] 与所述至少一个处理器通信连接的存储器;其中,

[0068] 所述存储器存储有可被所述至少一个处理器执行的指令,所述指令被所述至少一个处理器执行,以使所述至少一个处理器能够:

[0069] 若主节点从第一ODL节点切换为第二ODL节点,所述第二ODL节点基于自身的第二IP地址,通过所述networking-odl插件建立与Neutron Server服务器之间的初始化websocket连接;其中,所述第一ODL节点基于自身的第一IP地址,通过networking-odl插件建立与Neutron Server服务器之间的初始化websocket连接;通过所述networking-odl插件,创建关联的第一数据库监听器;断开与所述Neutron Server服务器之间的websocket连接;

[0070] 通过所述networking-odl插件,创建关联的第二数据库监听器;

[0071] 根据所述第二数据库监听器,从MDSAL数据库中监听到虚拟组网环境的业务状态变化数据,将所述业务状态变化数据通过所述websocket连接上报至所述Neutron Server服务器。

[0072] 本申请的一些实施例提供的一种ODL控制器的业务状态上报非易失性计算机存储介质,存储有计算机可执行指令,所述计算机可执行指令设置为:

[0073] 若主节点从第一ODL节点切换为第二ODL节点,所述第二ODL节点基于自身的第二IP地址,通过所述networking-odl插件建立与Neutron Server服务器之间的初始化websocket连接;其中,所述第一ODL节点基于自身的第一IP地址,通过networking-odl插件建立与Neutron Server服务器之间的初始化websocket连接;通过所述networking-odl插件,创建关联的第一数据库监听器;断开与所述Neutron Server服务器之间的websocket连接;

[0074] 通过所述networking-odl插件,创建关联的第二数据库监听器;

[0075] 根据所述第二数据库监听器,从MDSAL数据库中监听到虚拟组网环境的业务状态变化数据,将所述业务状态变化数据通过所述websocket连接上报至所述Neutron Server服务器。

[0076] 本申请中的各个实施例均采用递进的方式描述,各个实施例之间相同相似的部分互相参见即可,每个实施例重点说明的都是与其他实施例的不同之处。尤其,对于设备和介质实施例而言,由于其基本相似于方法实施例,所以描述的比较简单,相关之处参见方法实施例的部分说明即可。

[0077] 本申请实施例提供的设备和介质与方法是一一对应的,因此,设备和介质也具有与其对应的方法类似的有益技术效果,由于上面已经对方法的有益技术效果进行了详细说明,因此,这里不再赘述设备和介质的有益技术效果。

[0078] 本领域内的技术人员应明白,本发明的实施例可提供为方法、系统、或计算机程序产品。因此,本发明可采用完全硬件实施例、完全软件实施例、或结合软件和硬件方面的实施例的形式。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例的方法、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例而已,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。凡在本申请技术原理之内所作的任何修改、等同替换、改进等,均应落入本申请的保护范围之内。

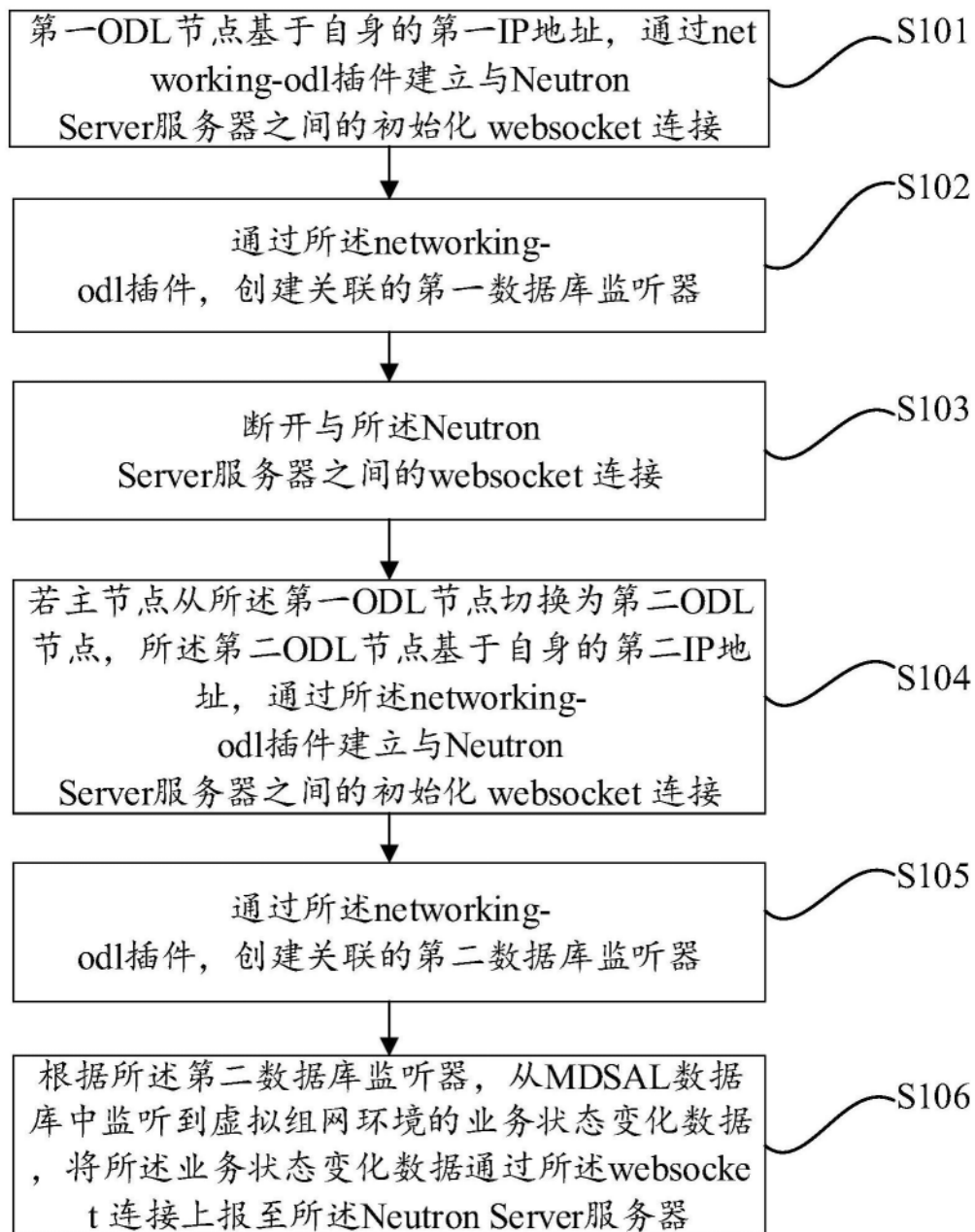


图1

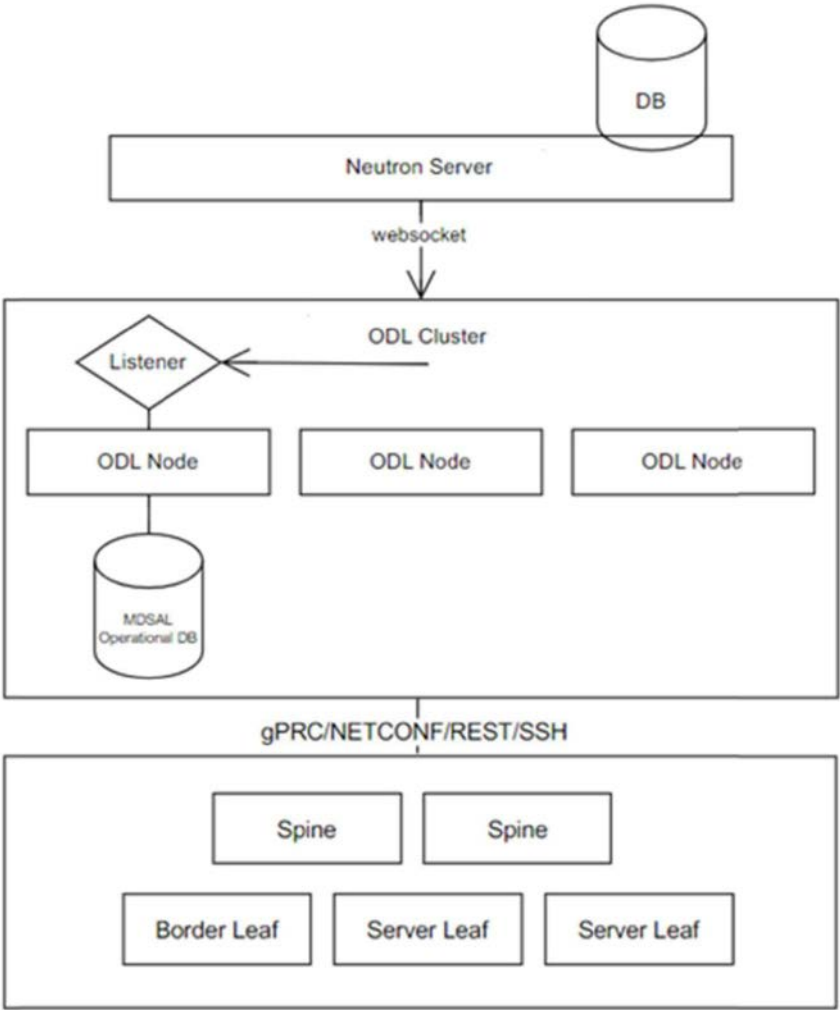


图2

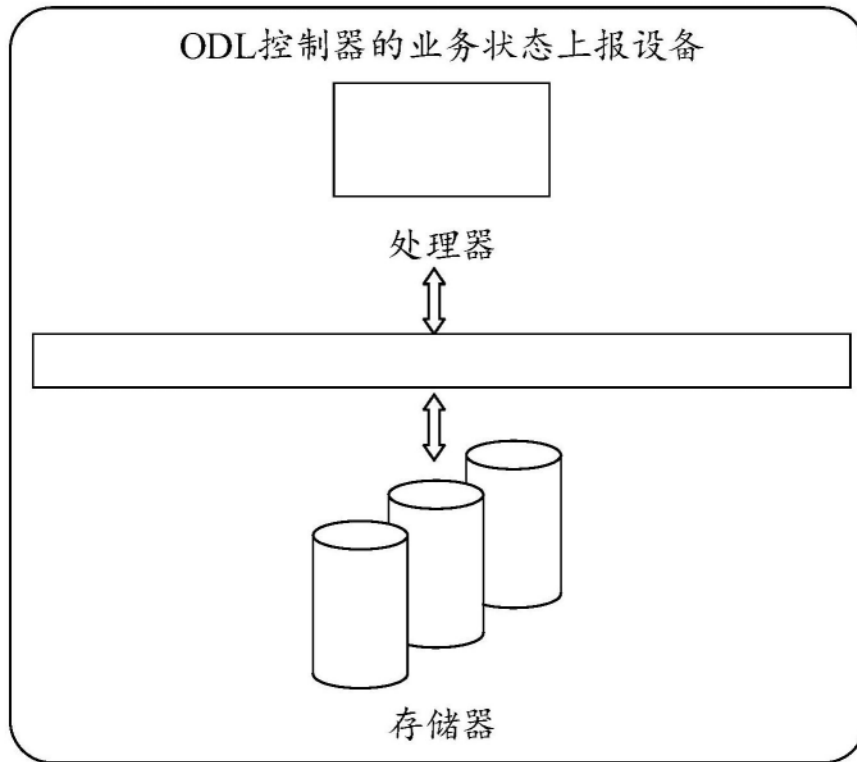


图3