(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 105049536 A
(43)申请公布日 2015.11.11

(21)申请号 201510567042.8

(22)申请日 2015.09.08

(71)申请人 南京大学
地址 210093 江苏省南京市鼓楼区汉口路22号

(72)发明人 孔繁宇 钱柱中 陆桑璐

(74)专利代理机构 南京苏高专利商标事务所（普通合伙） 32204
代理人 许丹丹

(51)Int.Cl.
*H04L 29/08*(2006.01)

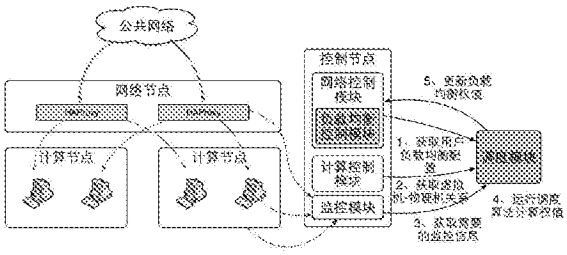
权利要求书2页 说明书6页 附图4页

(54)发明名称

IaaS云环境中的负载均衡系统和负载均衡方法

(57)摘要

本发明公开了一种IaaS云环境中的负载均衡系统和负载均衡方法，通过增加负载均衡调度模块，实现对负载均衡器的周期性调整，解决了IaaS云环境中现有的负载均衡器虽然可以对流量进行分发以及对后端机器进行负载均衡，但是由于负载均衡器的后端机器由传统的物理机变成了虚拟机，而负载均衡器并没有办法针对虚拟机和物理机的对应关系做出合适的负载均衡策略从而导致的云环境中物理机负载不均衡的问题，本发明的负载均衡系统和负载均衡方法针对上述情况，通过结合云环境拓扑及实时监控信息指导负载均衡器进行周期性的权重调整，达到平衡云环境中物理机负载的效果。




CN 105049536 A

1. 一种 IaaS 云环境中的负载均衡系统，包括：网络节点、计算节点以及控制节点，网络节点上运行有负载均衡器，所述负载均衡器负责对网络流量进行分发；控制节点上运行有网络控制模块、负载均衡控制模块、计算控制模块以及监控模块；其特征在于，该系统还包括负载均衡调度模块，所述负载均衡调度模块与所述控制节点进行周期性信息交互并通过所述控制节点对所述负载均衡器的权重分配进行调整。

2. 根据权利要求 1 所述的 IaaS 云环境中的负载均衡系统，其特征在于，所述负载均衡调度模块与所述控制节点进行周期性信息交互包括如下步骤：

(1) 所述负载均衡调度模块与所述网络控制模块进行交互，获取用户负载均衡配置信息，所述用户负载均衡配置信息包括所有用户配置的所有负载均衡池的信息；

(2) 所述负载均衡调度模块与所述计算控制模块进行交互，获取虚拟机与物理机的对应关系，并与步骤 (1) 中获得的负载均衡池中的虚拟机信息进行关联，形成拓扑关系；

(3) 所述负载均衡调度模块与所述监控模块进行交互，获取需要的监控信息，所述监控信息为物理机以及虚拟机的实时 CPU 占用率和 I/O 负载监控数据；

(4) 以全部物理机的最大 CPU 占用率取最小值为优化目标建立线性规划问题，并进行求解得到所述线性规划问题的最优解；

(5) 将步骤 (4) 中的最优解转换为各个负载均衡池中虚拟机的权重；

(6) 所述负载均衡调度模块将所述各个负载均衡池中虚拟机的权重信息反馈给所述网络控制模块请求修改所述负载均衡器的权重分配；

(7) 所述控制节点负责调整所述负载均衡器的后端权重，完成一次负载均衡。

3. 根据权利要求 2 所述的 IaaS 云环境中的负载均衡系统，其特征在于，步骤 (4) 中建立的线性规划问题如下：

优化目标：min T

约束条件：

1) 对于每个负载均衡池 $P_i$，有：$\sum_{v_j \in P_i} l_j = \sum_{v_j \in P_i} l'_j$

2) 对于每台虚拟机 $v_j$，有：$l'_j \leq c_j$

3) 对于每台物理机 $h_i$，有：$L'_i = \sum_{v_j \in S(h_i)} l'_j \leq C_i$

4) 对于每台物理机 $h_i$，有：$\frac{L'_i}{C_i} \leq T$

式中，T 为物理机的 CPU 占用率，$C_i$ 为物理机 $h_i$ 的最大容量；$l_j$ 为负载均衡池 $P_i$ 中虚拟机 $v_j$ 均衡调整前的负载，$l'_j$ 为虚拟机 $v_j$ 均衡调整后的负载，$c_j$ 为虚拟机的最大容量；对于每一个物理机 $h_i$，$L'_i$ 为物理机 $h_i$ 均衡调整后的负载，$S(h_i)$ 为物理机 $h_i$ 上的虚拟机集合。

4. 一种 IaaS 云环境中的负载均衡方法，其特征在于，负载均衡调度模块与控制节点上运行的网络控制模块、计算控制模块、监控模块进行周期性信息交互并通过所述控制节点对负载均衡器的权重分配进行调整实现负载均衡，进行一次所述负载均衡包括以下步骤：

(1) 所述负载均衡调度模块与所述网络控制模块进行交互，获取用户负载均衡配置信息，所述用户负载均衡配置信息包括所有用户配置的所有负载均衡池的信息；

(2) 所述负载均衡调度模块与所述计算控制模块进行交互，获取虚拟机与物理机的对

应关系，并与步骤 (1) 中获得的负载均衡池中的虚拟机信息进行关联，形成拓扑关系；

(3) 所述负载均衡调度模块与所述监控模块进行交互，获取需要的监控信息，所述监控信息为物理机以及虚拟机的实时 CPU 占用率和 I/O 负载监控数据；

(4) 以全部物理机的最大 CPU 占用率取最小值为优化目标建立线性规划问题，并进行求解得到所述线性规划问题的最优解；

(5) 将步骤 (4) 中的最优解转换为各个负载均衡池中虚拟机的权重；

(6) 所述负载均衡调度模块将所述各个负载均衡池中虚拟机的权重信息回馈给所述网络控制模块请求修改所述负载均衡器的权重分配；

(7) 所述控制节点负责调整所述负载均衡器的后端权重。

5. 根据权利要求 4 所述的 IaaS 云环境中的负载均衡方法，其特征在于，步骤 (4) 中建立的所述线性规划问题如下：

优化目标 :min T

约束条件：

1) 对于每个负载均衡池 $P_i$，有：$\sum_{v_j \in P_i} l_j = \sum_{v_j \in P_i} l_j'$

2) 对于每台虚拟机 $v_j$，有：$l'_j \leq c_j$

3) 对于每台物理机 $h_i$，有：$L_i' = \sum_{v_j \in S(h_i)} l_j' \leq C_i$

4) 对于每台物理机 $h_i$，有：$\frac{L_i'}{C_i} \leq T$

式中，T 为物理机的 CPU 占用率，$C_i$ 为物理机 $h_i$ 的最大容量；$l_j$ 为负载均衡池 $P_i$ 中虚拟机 $v_j$ 均衡调整前的负载，$l'_j$ 为虚拟机 $v_j$ 均衡调整后的负载，$c_j$ 为虚拟机的最大容量；对于每一个物理机 $h_i$，$L'_i$ 为物理机 $h_i$ 均衡调整后的负载，$S(h_i)$ 为物理机 $h_i$ 上的虚拟机集合。

# IaaS 云环境中的负载均衡系统和负载均衡方法

## 技术领域

[0001] 本发明涉及云计算领域中的负载均衡技术，尤其涉及一种 IaaS 云环境中的负载均衡系统和负载均衡方法。

## 背景技术

[0002] 基础设施即服务 IaaS 所提供给消费者的服务是对所有计算基础设施的利用，包括处理 CPU、内存、存储、网络和其它基本的计算资源，用户能够部署和运行任意软件，包括操作系统和应用程序。消费者不管理或控制任何云计算基础设施，但能控制操作系统的选择、存储空间、部署的应用，也有可能获得有限制的网络组件（例如路由器、防火墙、负载均衡器等）的控制。

[0003] 负载均衡器是一种把网络请求分散到一个服务器集群中的可用服务器上去，通过管理进入的 Web 数据流量和增加有效的网络带宽。当前负载均衡技术广泛应用于网络服务器中 (2-4 层网络)，横向拓展的特性非常适合于网络服务器对于负载压力的应对，各类负载均衡方案层出不穷，如硬件的 F5、软件的 LVS、HAProxy、Nginx 等，在长期的实践中已经成为了主流的系统架构设计方法。

[0004] OpenStack 是一个开源的云计算管理平台项目，由几个主要的组件组合起来完成具体工作。OpenStack 通过各种互补的服务提供了基础设施即服务 IaaS 的解决方案，每个服务提供 API 以进行集成。

[0005] 当传统应用迁移到 IaaS 云环境中后，由于以下原因，仍然需要采用对虚拟机之间进行负载均衡来达到横向拓展：

[0006] (1) 大量已经编写好的应用在系统设计架构上基于负载均衡而设计，非常适合于做横向拓展；(2) 纵向拓展（如创建性能更强的虚拟机）由于不能超过物理服务器本身的性能极限，所以并不能满足大型应用的需求；(3) 横向拓展避免了应用可能出现的单点故障问题，增强了系统整体鲁棒性，有利于应用的稳定运行。

[0007] 前两点基于应用的可拓展性 (Scalability) 要求，第三点基于应用的可用性 (Availability) 要求，因此，在云环境中，云服务提供商（如 OpenStack）本身为用户提供了负载均衡服务 (LBaaS)，用户可以通过简单的配置在自己创建的虚拟机之间建立负载均衡。

[0008] 但是由于云环境和传统的物理机环境不同，直接在虚拟机之间建立的负载均衡关系的做法由于无视了底层的物理服务器关系，虽然虚拟机之间维持了均衡的负载，但是非常容易造成底层的物理服务器之间的负载严重不平衡，而较高的物理机负载又会反过来影响虚拟机的性能，造成整体云服务性能的下降。

## 发明内容

[0009] 发明目的：为了解决现有技术中存在的问题，本发明提供了一种 IaaS 云环境中的负载均衡系统和负载均衡方法，基于整体云环境的结构信息和监控信息（主要包含 CPU 利用率和 I/O 负载信息），对负载均衡的策略进行调整，通过对虚拟机之间负载均衡的比例进

行调整，来平衡整体物理机的CPU利用率及I/O负载，从而达到提升整体云服务性能、优化云环境整体效率的效果。

[0010] 技术方案：本发明的IaaS云环境中的负载均衡系统包括：网络节点、计算节点以及控制节点，网络节点上运行有负载均衡器，所述负载均衡器负责对网络流量进行分发；控制节点上运行有网络控制模块、负载均衡控制模块、计算控制模块以及监控模块；其特征在于，该系统还包括负载均衡调度模块，所述负载均衡调度模块与所述控制节点进行周期性信息交互并通过所述控制节点对所述负载均衡器的权重分配进行调整。

[0011] 其中，所述负载均衡调度模块与所述控制节点进行周期性信息交互具体如下：

[0012] (1)所述负载均衡调度模块与所述网络控制模块进行交互，获取用户负载均衡配置信息，所述用户负载均衡配置信息包括所有用户配置的所有负载均衡池的信息；

[0013] (2)所述负载均衡调度模块与所述计算控制模块进行交互，获取虚拟机与物理机的对应关系，并与步骤(1)中获得的负载均衡池中的虚拟机信息进行关联，形成拓扑关系；

[0014] (3)所述负载均衡调度模块与所述监控模块进行交互，获取需要的监控信息，所述监控信息为物理机以及虚拟机的实时CPU占用率和I/O负载监控数据；

[0015] (4)以全部物理机的最大CPU占用率取最小值为优化目标建立线性规划问题，并进行求解得到所述线性规划问题的最优解；

[0016] (5)将步骤(4)中的最优解转换为各个负载均衡池中虚拟机的权重；

[0017] (6)所述负载均衡调度模块将所述各个负载均衡池中虚拟机的权重信息反馈给所述网络控制模块请求修改所述负载均衡器的权重分配；

[0018] (7)所述控制节点负责调整所述负载均衡器的后端权重，完成一次负载均衡。

[0019] 其中，步骤(4)中建立的线性规划问题如下：

[0020] 优化目标：min T

[0021] 约束条件：

[0022] 1)对于每个负载均衡池$P_i$，有：$\sum_{v_j \in P_i} l_j = \sum_{v_j \in P_i} l_j'$

[0023] 2)对于每台虚拟机$v_j$，有：$l_j' \le c_j$

[0024] 3)对于每台物理机$h_i$，有：$L_i' = \sum_{v_j \in S(h_i)} l_j' \le C_i$

[0025] 4)对于每台物理机$h_i$，有：$\frac{L_i'}{C_i} \le T$

[0026] 式中，T为物理机的CPU占用率，$C_i$为物理机$h_i$的最大容量；$l_j$为负载均衡池$P_i$中虚拟机$v_j$均衡调整前的负载，$l'_j$为虚拟机$v_j$均衡调整后的负载，$c_j$为虚拟机的最大容量；对于每一个物理机$h_i$，$L'_i$为物理机$h_i$均衡调整后的负载，$S(h_i)$为物理机$h_i$上的虚拟机集合。

[0027] 本发明还公开了一种IaaS云环境中的负载均衡方法，负载均衡调度模块与控制节点上运行的网络控制模块、计算控制模块、监控模块进行周期性信息交互并通过所述控制节点对负载均衡器的权重分配进行调整实现负载均衡，进行一次所述负载均衡包括以下步骤：

[0028] (1)所述负载均衡调度模块与所述网络控制模块进行交互，获取用户负载均衡配置信息，所述用户负载均衡配置信息包括所有用户配置的所有负载均衡池的信息；

[0029] (2) 所述负载均衡调度模块与所述计算控制模块进行交互，获取虚拟机与物理机的对应关系，并与步骤 (1) 中获得的负载均衡池中的虚拟机信息进行关联，形成拓扑关系；

[0030] (3) 所述负载均衡调度模块与所述监控模块进行交互，获取需要的监控信息，所述监控信息为物理机以及虚拟机的实时 CPU 占用率和 I/O 负载监控数据；

[0031] (4) 以全部物理机的最大 CPU 占用率取最小值为优化目标建立线性规划问题，并进行求解得到所述线性规划问题的最优解；

[0032] (5) 将步骤 (4) 中的最优解转换为各个负载均衡池中虚拟机的权重；

[0033] (6) 所述负载均衡调度模块将所述各个负载均衡池中虚拟机的权重信息回馈给所述网络控制模块请求修改所述负载均衡器的权重分配；

[0034] (7) 所述控制节点负责调整所述负载均衡器的后端权重。

[0035] 其中，步骤 (4) 中建立的所述线性规划问题如下：

[0036] 优化目标：min T

[0037] 约束条件：

[0038] 1) 对于每个负载均衡池 $P_i$，有：$\sum_{v_j \in P_i} l_j = \sum_{v_j \in P_i} l'_j$

[0039] 2) 对于每台虚拟机 $v_j$，有：$l'_j \le c_j$

[0040] 3) 对于每台物理机 $h_i$，有：$L'_i = \sum_{v_j \in S(h_i)} l'_j \le C_i$

[0041] 4) 对于每台物理机 $h_i$，有：$\frac{L'_i}{C_i} \le T$

[0042] 式中，T 为物理机的 CPU 占用率，$C_i$ 为物理机 $h_i$ 的最大容量；$l_j$ 为负载均衡池 $P_i$ 中虚拟机 $v_j$ 均衡调整前的负载，$l'_j$ 为虚拟机 $v_j$ 均衡调整后的负载，$c_j$ 为虚拟机的最大容量；对于每一个物理机 $h_i$，$L'_i$ 为物理机 $h_i$ 均衡调整后的负载，$S(h_i)$ 为物理机 $h_i$ 上的虚拟机集合。

[0043] 有益效果：本发明提出了基于线性规划的云环境负载均衡，通过结合云环境的拓扑及实时监控信息指导负载均衡器权值的调整，以达到优化云环境整体效率的目的。负载均衡调度模块基于整体云环境的结构信息和监控信息，对负载均衡的策略进行调整，通过对虚拟机之间负载均衡的比例进行调整，来平衡整体物理机的负载，从而达到提升整体云服务性能的目的。

## 附图说明

[0044] 图 1 是结合 OpenStack 平台的负载均衡系统的结构示意图；

[0045] 图 2 是图 1 中的负载均衡系统收集数据示意图；

[0046] 图 3 是虚拟机和物理机的拓扑结构示意图；

[0047] 图 4 是虚拟机 v1 的 CPU 负载变化监控图；

[0048] 图 5 是虚拟机 v2 的 CPU 负载变化监控图；

[0049] 图 6 是虚拟机 v3 的 CPU 负载变化监控图；

[0050] 图 7 是物理机 h1 的 CPU 负载变化监控图；

[0051] 图 8 是物理机 h2 的 CPU 负载变化监控图。

## 具体实施方式

[0052] 为了便于理解，下面结合实施例与附图对本发明作进一步的说明，实施方式提及的内容并非对本发明的限定。

[0053] 图 1 以 OpenStack 平台为例介绍本发明的负载均衡系统，图中左侧部分为 OpenStack 的网络布局，包含了连接公共网络的网络节点、计算节点以及标准的控制节点；网络节点负责为租户提供虚拟网络服务，主要专注于租户虚拟网络的实现；计算节点负责为用户提供计算资源，通过运行用户指定的虚拟机将计算资源提供给用户，控制节点为整个云环境的中央管理节点，负责云平台内部的管理和通信，并且作为用户的控制接口。

[0054] 用户对负载均衡池的访问请求通过网络节点进行，在网络节点上运行有 HAProxy 模块，HAProxy 模块作为负载均衡器，负责对网络流量进行分发；控制节点上同时运行有计算控制模块 Nova、网络控制模块 Neutron、负载均衡控制模块 LBaaS 和监控模块 Ceilometer；计算控制模块用来控制用户的虚拟机在计算节点上的运行；网络控制模块用来控制用户在虚拟机上建立的虚拟网络服务；负载均衡控制模块用来控制用户创建的虚拟机负载均衡服务；监控模块用于收集云环境中的实时监控信息，包括计算节点上运行的虚拟机以及真实计算节点对应的物理机的 CPU、I/O 等各个方面的实时负载信息等；图 1 展示了上述模块的基本结构，可以看到控制节点在整个云环境中通过计算、网络、负载均衡控制模块的共同协同工作，将运行在各个计算机节点上的用户虚拟机组织成虚拟网络，并在此基础上建立负载均衡服务。其中右侧为负载均衡调度模块，该模块可以把网络请求分散到一个服务器集群中的可用服务器上去，通过管理进入的 Web 数据流量和增加有效的网络带宽，并且支持在运行过程中对处于负载均衡状态下的后端机器进行动态的权重调整。负载均衡调度模块周期性运行对云环境负载均衡做出调整，其一次运行过程如下：

[0055] (1) 使用 OpenStack 提供的 neutronclient 或其他方法访问网络控制模块 API，与 OpenStack 的网络控制模块进行交互，调用 Listpools API(/v2.0/lbaas/pools)，获取用户负载均衡配置信息，即：全局所有用户配置的所有负载均衡池的信息；

[0056] (2) 使用 OpenStack 提供的 novaclient 或其他方法访问计算控制模块 API，与 OpenStack 的计算控制模块进行交互，调用 List server detailsAPI(/v2.1/servers/detail) 获取云环境中虚拟机与物理机的逻辑对应关系，即某台虚拟机实际运行在云环境中具体哪一台计算节点上，并与上一步中获得的负载均衡池中的虚拟机信息进行连接，形成如图 2 所示的拓扑关系；

[0057] (3) 使用 OpenStack 提供的 ceilometerclient 或其他方法访问监控模块 API，与 OpenStack 的监控模块进行交互，调用 List samplesAPI(/v2/samples)，获得需要的监控信息，即：物理机以及虚拟机的实时 CPU 利用率及 I/O 负载监控数据，如图 2 左侧列出的监控项所示；

[0058] (4) 建立线性规划问题，并进行求解得到最优解；

[0059] (5) 将步骤 (4) 中的最优解转换为各个负载均衡池中虚拟机的权重；

[0060] (6) 将新的权重信息通过 OpenStack 提供的 neutronclient 或其他方法访问网络控制模块 API 反馈给 OpenStack 网络控制模块，对于调整后的每一台虚拟机，调用 Update pool memberAPI(/v2.0/lbaas/pools/{pool_id}/members/{member_id}) 请求修改负载均衡器权值；

[0061] (7)OpenStack的控制节点在执行API调用后自动修改对应HAProxy的配置文件完成HAProxy后端权重的调整,完成一次负载均衡。

[0062] OpenStack中默认通过HAProxy提供软件负载均衡功能,用户可以通过创建负载均衡池并将想要作为负载均衡后端的虚拟机绑定到该负载均衡来建立负载均衡服务。创建成功后通过负载均衡池所绑定的公共IP即可访问所有虚拟机后端构成的负载均衡集群。

[0063] HAProxy的一个重要功能在于可以实时的调整负载均衡后端机器的比例权重,实时修改后端虚拟机的负载分配比例。

[0064] 本发明中将物理机的CPU利用率和I/O负载作为首要的负载衡量标准,上述步骤(4)中所建立的线性规划问题具体如下:

[0065] 问题输入:

[0066] 1)共有N台物理机(计算节点),物理机$h_i$的负载为$L_i$,每台物理机的最大容量为$C_i$;

[0067] 2)系统中虚拟机$v_j$的负载为$l_j$,最大容量为$c_j$;

[0068] 3)共有M个负载均衡池,每个负载均衡池$P_i$对应一组虚拟机:$P_i=\{v_1, v_2, v_3, \cdots v_j \cdots\}$,且每个虚拟机$v_j$仅能存在于一个负载均衡池中,即对于任意两个负载均衡池,其交集为空集;

[0069] 4)对于每一个物理机$h_i$,$S(h_i)$表示这台物理机上的虚拟机集合;

[0070] 问题输出:对于每个负载均衡池$P_i$,对所有的虚拟机$v_j \in P_i$,给出新的期望负载$l'_j$;

[0071] 求解该问题后得到期望负载,将其转换为比例后即可用来指导HAProxy进行各个后端虚拟机的动态权重调整,以达到云环境整体负载优化的目的。求解上述问题的线性规划问题,具体如下:

[0072] 约束条件:

[0073] 1)对于每个负载均衡池$P_i$,其总负载不变:$\sum_{v_j \in P_i} l_j = \sum_{v_j \in P_i} l'_j$;

[0074] 2)对于每台虚拟机$v_j$,其负载不超过最大容量:$l'_j \le c_j$;

[0075] 3)对于每台物理机$h_i$,其负载不超过最大容量:$\sum_{v_j \in S(h_i)} l'_j \le C_i$;

[0076] 优化目标:添加变量T,对所有物理机$h_i$,有$\frac{\sum_{v_j \in S(h_i)} l'_j}{C_i} \le T$,规划目标为min T。

[0077] 通过求解该线性规划问题,即可得到先前定义问题的解。

[0078] 以下结合具体数值对本发明所提供的IaaS云环境中的负载均衡方法做更进一步详细介绍。

[0079] 用户设置了两个负载均衡池P1和P2,P1对应于虚拟机{v1,v2},P2对应于虚拟机{v3},虚拟机v1运行在物理机h1上,虚拟机v2,v3运行在物理机h2上。此处作为决策依据的负载可以采用CPU占用率和I/O负载,为便于理解,此处取CPU占用率作为负载衡量指标,虚拟机均为2CPU机器,物理机均为4CPU机器。所有虚拟机上都运行着相同的产生CPU负载的Web应用,当收到用户请求后进行一定量的CPU运算后返回。

[0080] 通过向负载均衡池P1产生两倍于P2的访问量,使得所有虚拟机负载均为50%,生

成虚拟机负载如下（机器负载＝实时 CPU 利用率＊机器 CPU 核数量）：

[0081] $$\begin{cases} l_1 = 1 \\ l_2 = 1 \\ l_3 = 1 \end{cases}$$

[0082] 对应的物理机负载为：

[0083] $$\begin{cases} L_1 = 1 \\ L_2 = 2 \end{cases}$$

[0084] 1. 建立线性规划问题并进行运行算法求解

[0085] 在本实施例中根据本发明中描述的问题及算法可以得到对应的线性规划问题：

[0086] $$\begin{cases} l_1' + l_2' = l_1 + l_2 = 2 \\ l_3' = l_3 = 1 \\ l_1' \le c_1 = 2 \\ l_2' \le c_2 = 2 \\ l_3' \le c_3 = 2 \\ l_1' \le C_1 = 4 \\ l_2' + l_3' \le C_2 = 4 \\ \dfrac{l_1'}{C_1} = \dfrac{l_1'}{4} \le T \\ \dfrac{l_2' + l_3'}{C_2} = \dfrac{l_2' + l_3'}{4} \le T \end{cases}$$

[0087] 采用单纯形法求解该问题，得到结果：

[0088] $$\begin{cases} l_1' = 1.5 \\ l_2' = 0.5 \\ l_3' = 1 \end{cases}$$

[0089] 2. 调整权重

[0090] 根据上一步中得到的结果，可以确定负载均衡池 P1 中新的权重比例关系应该修改为 3:1，使用此比例对 HAProxy 进行调整。

[0091] 3. 实验测试效果

[0092] 实验的效果如图 4 至图 8 所示，图 4、图 5 以及图 6 中可以看到在图的中间靠右位置（约 10:54:45 的位置），在运行本发明的负载均衡方法后，v1 的负载出现了上涨而 v2 的负载出现了下降，v3 则保持不变。图 7 和图 8 中可以看出在运行本发明的负载均衡方法后，两台物理机的权重发生了改变，并趋向于平衡。

[0093] 本发明具体应用途径很多，上述以 OpenStack 平台为例进行介绍，仅为便于理解设置，本发明中的负载均衡调度模块以及负载均衡方法对于支持负载均衡功能的 IaaS 云环境的如 OpenNebula(HTTP 的 load balancer 模块）、CloudStack(OVS Plugin 提供 Load Balancing 功能）和 OpenStack(LBaaS 模块）等开源平台均适用，应当指出以上实施例对本发明不构成限定，相关工作人员在不偏离本发明技术思想的范围内，所进行的多样变化和修改，均落在本发明的保护范围内。

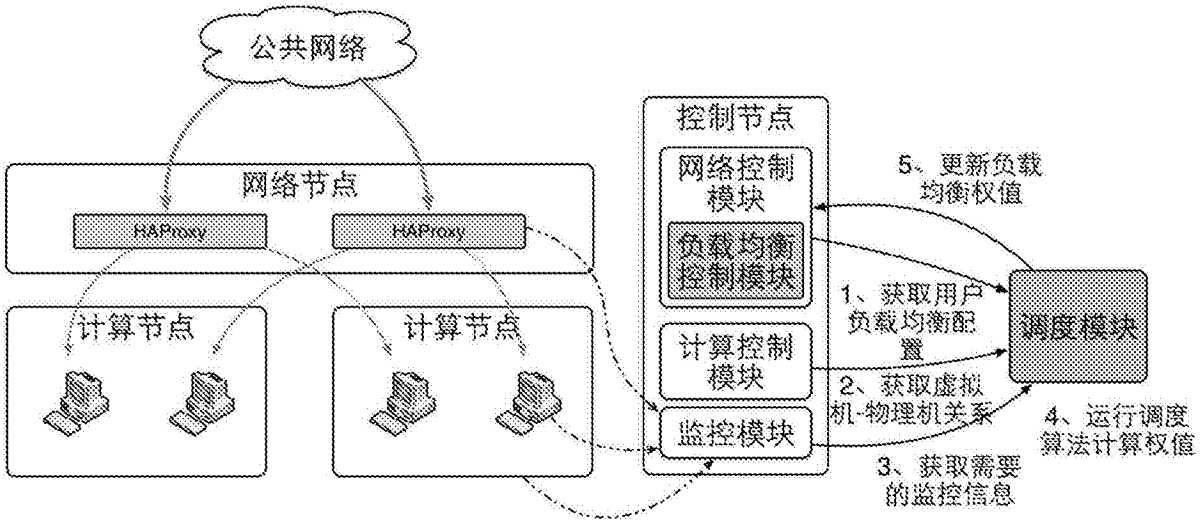

公共网络
网络节点
HAProxy
HAProxy
计算节点
计算节点
控制节点
网络控制模块
负载均衡控制模块
计算控制模块
监控模块
5、更新负载均衡权值
1、获取用户负载均衡配置
2、获取虚拟机-物理机关系
3、获取需要的监控信息
调度模块
4、运行调度算法计算权值

图 1

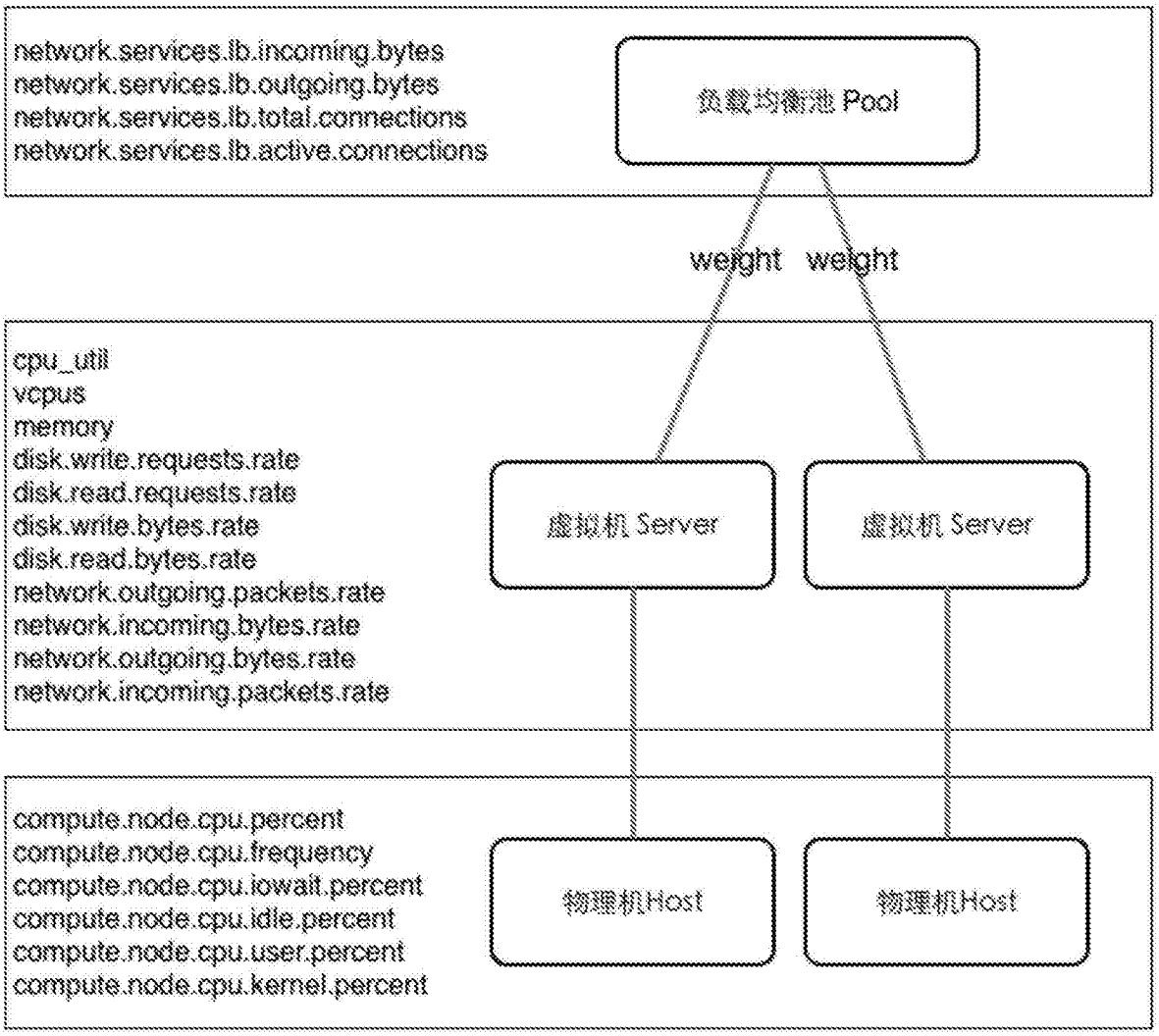

network.services.lb.incoming.bytes
network.services.lb.outgoing.bytes
network.services.lb.total.connections
network.services.lb.active.connections
负载均衡池 Pool
weight
weight
cpu_util
vcpus
memory
disk.write.requests.rate
disk.read.requests.rate
disk.write.bytes.rate
disk.read.bytes.rate
network.outgoing.packets.rate
network.incoming.bytes.rate
network.outgoing.bytes.rate
network.incoming.packets.rate
虚拟机 Server
虚拟机 Server
compute.node.cpu.percent
compute.node.cpu.frequency
compute.node.cpu.iowait.percent
compute.node.cpu.idle.percent
compute.node.cpu.user.percent
compute.node.cpu.kernel.percent
物理机Host
物理机Host

图 2

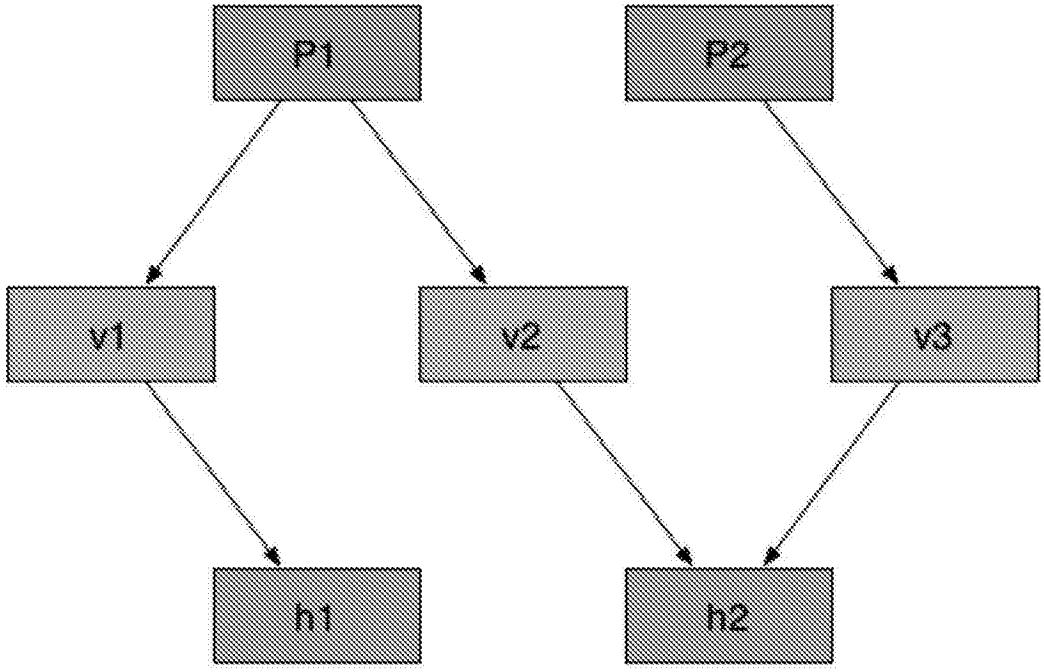



图 3

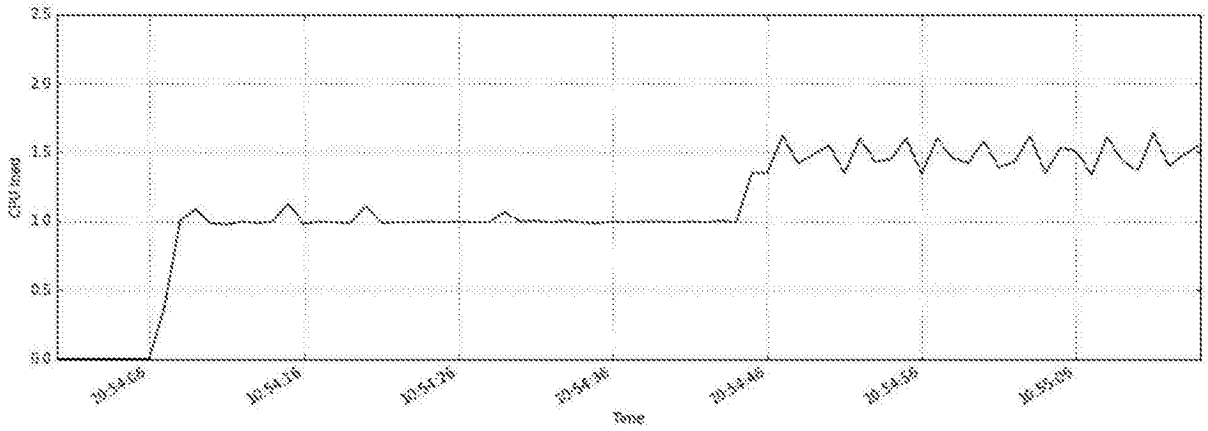



图 4

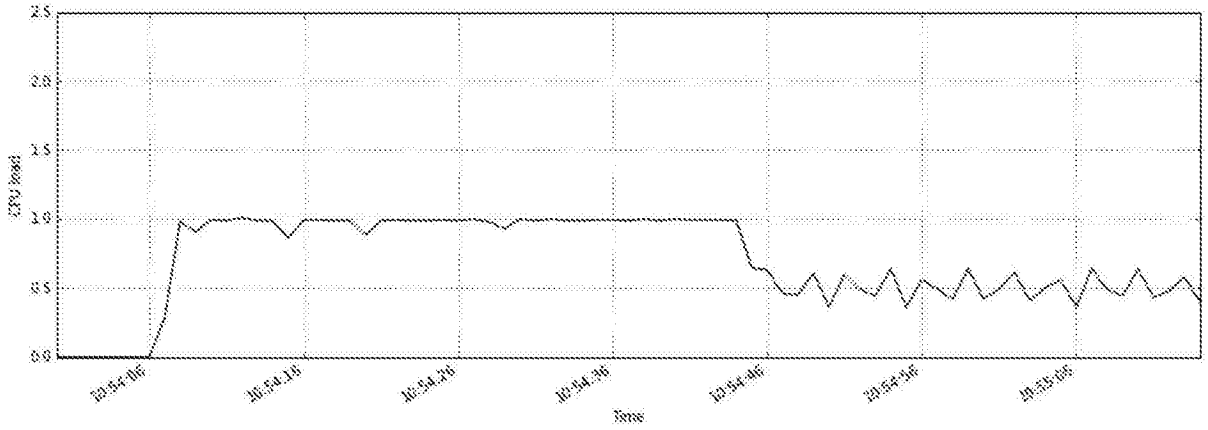

2.5
2.0
1.5
1.0
0.5
0.0
CPU load
Time

图 5

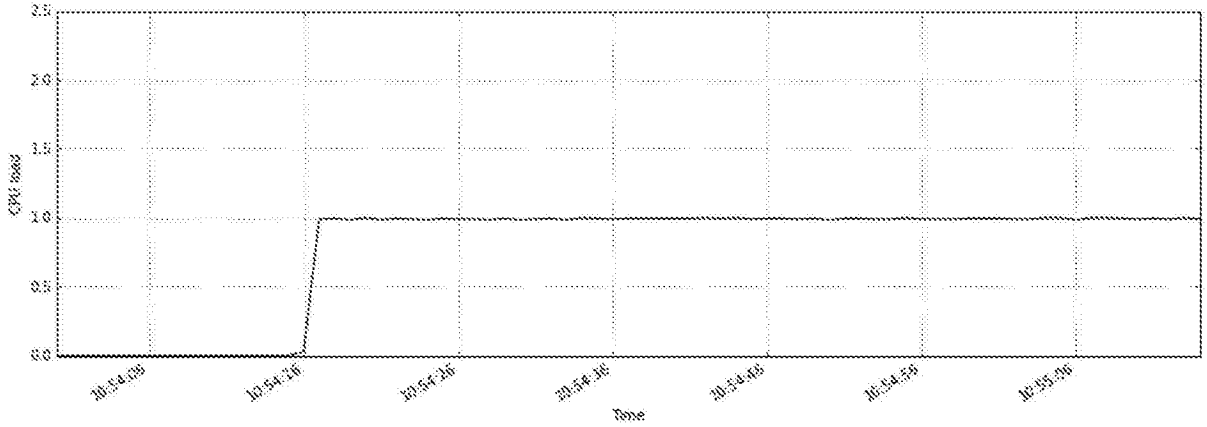

2.5
2.0
1.5
1.0
0.5
0.0
CPU load
Time

图 6

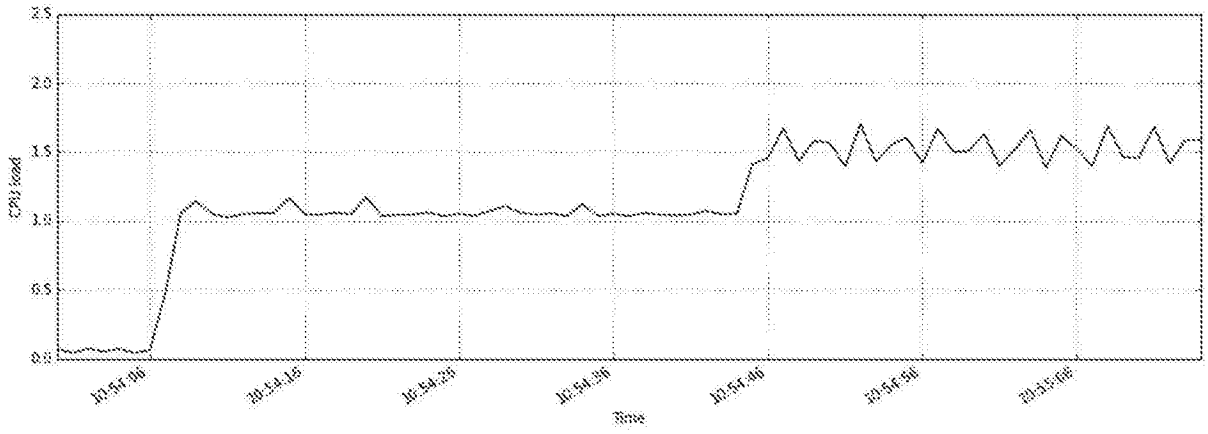

2.5
2.0
1.5
1.0
0.5
0.0
CPU load
Time

图 7

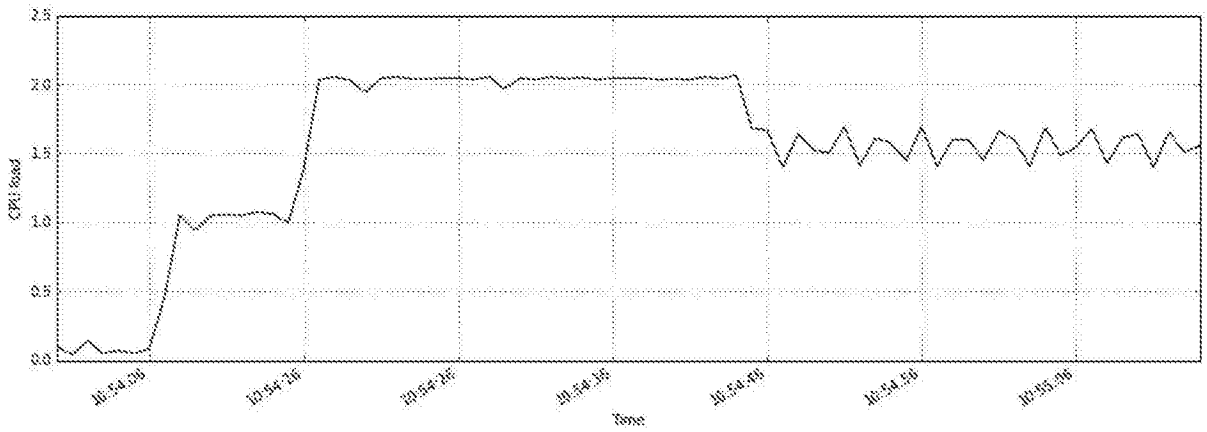



图 8