

### 参赛队伍信息表

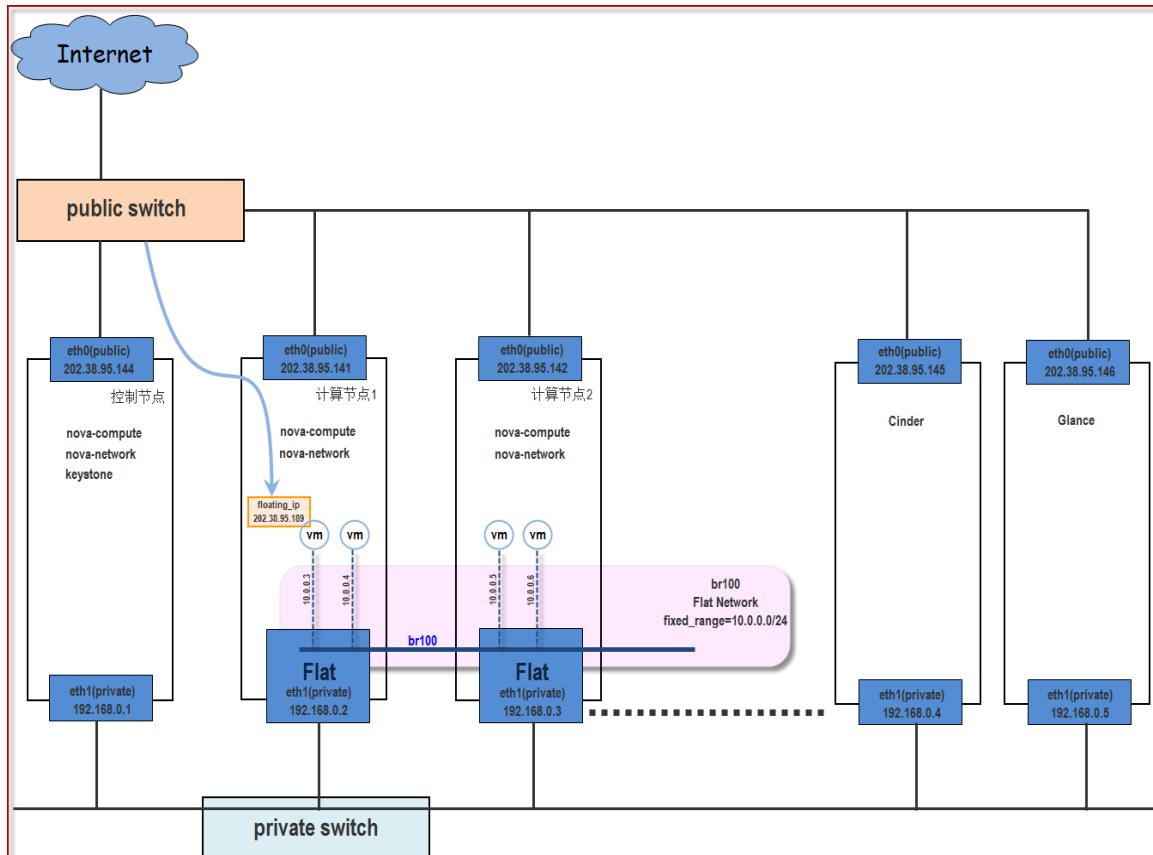
队伍名称	幻影之翼		
领队姓名	疏官胜	指导老师	张焕杰
邮箱地址	sgs2012@mail.ustc.ed u.cn	领队电话	15255126516
所属学校 (单位)	中国科学技术大学		
队员一	王硕	队员二	张俊
队员三	徐华	队员四	
队员五		队员六	

应用名称	云数据中心业务流量 QoS 保障
<b>一、应用的简介和摘要（研究内容、解决问题等，600 字以内）</b>	
<p>随着云计算技术及其业务的普及，数据中心云化发展逐步走向大型化、自动化、虚拟化，越来越多的分布式数据密集型应用倾向于部署在云环境中，如 MapReduce、Storm、Spark 等，云网络中产生超大规模东西向数据流量已经成为必然趋势。上层应用共享底层物理网络带宽，网络拥塞成为常见现象，云业务服务质量需要保障。</p>	
<p>云基础设施本质是一个虚拟化资源池，单个操作系统和网络端口不再一一对应，同一台物理服务器上运行着的多台虚拟机对物理网卡存在竞争。传统网络 QoS 策略只针对物理端口，服务器所有网络行为由接入交换机直接控制。而在虚拟化环境中，虚拟机和交换机间隔一层服务器网卡，虚拟机层面的业务流量难以区分、管理和控制，其通过预定义 SLA 作为 QoS 的参数，缺乏一定的灵活性，SDN 提供开放应用编程接口，为 QoS 保障带来新契机。</p>	
<p>本应用提出一套基于 SDN 的云数据中心业务流量 QoS 保障的解决方案，为云中虚拟机业务流量提供自动化 QoS 保障的同时，尽可能提高物理链路利用率。通过增加虚拟接入层软件交换机，将网络接入移至物理服务器内部，进而实现对虚拟机网络策略的管理。借助 SDN 南向接口，实现针对业务类型的个性化流表自动下发机制，达到虚拟机业务流量调度到特别端口队列的目的。借助 SDN 北向接口，监控全局虚拟机业务流量状态以及端口队列流量状态，并在此基础上设计基于实时业务流量端口队列带宽动态调整策略。与此同时，采取分级控制的思想，在端口队列针设计基于流的限速策略，从而达到虚拟机业务流量 QoS 保障的目的。</p>	

二、应用场景介绍（包括示意图，800字以内）

科大瀚海星云校园云在开源软件 OpenStack 基础上部署云计算 IaaS 层服务，共计 40 个计算节点，可提供 CPU 数目 880，内存大小 1760G，磁盘大小 80TB，已注册用户近 1000 余人，资源使用率处于超饱和状态。其运行二年有余，面向全体师生提供计算和存储服务，有效地支撑了本校科教活动。然而，校园云平台本身依然遭遇若干挑战，如云中虚拟机的流量管理问题。针对物理网卡相同宿主机中的虚拟机之间在带宽上存在竞争关系，如一台虚拟机使用 P2P 的下载方式将会对宿主机中其他虚拟机正常对外服务造成极大影响等。传统云中网络基于 Linux bridge 的方式虽然可以针对网桥接口实现基于业务流量的 QoS 保障，但现有的 bridge 模块因缺乏适合的应用编程接口不能快速应对动态变化迅速的云网环境，物理交换机无法识别和区分虚拟机的业务流量信息，因此无法在真正意义上解决虚拟接入的问题，从而无法对虚拟机的流量进行监控和管理，造成用户虚拟机在网络带宽上的 QoS 不能得到保证，这也是校园云能够继续提供稳定服务的主要障碍之一。

目前，科大瀚海星云校园云网络实现主要基于 Linux Bridge 的方式，在物理网卡上建立网桥，网络拓扑如下图所示：



(图 2-1 瀚海星云网络拓扑)

本应用拟对科大校园云网络架构重新设计，通过利用支持 openflow 协议的虚拟

交换机替代 Linux 桥接的方式，提供完全自由和开放的网络设备编程接口，进而完成虚拟接入功能，从而为虚拟机业务流量自动化 QoS 保障提供支撑。本应用在校园云中主要解决：

①重要应用带宽保障

校园云主要致力于学校科教活动，一些常见的与科研相关的业务如 MapReduce 等，其服务质量应优先被保障。

②异常大流量限速

对于校园云中非常规流量，如若长时间维持占用大流量带宽，则采取一定程度的限速措施。

③恶意流量屏蔽

科大校园云提供 IaaS 云服务，用户申请的虚拟云主机可能出现安全性问题，主要体现在发送流量攻击包或者做 SSH 端口扫描等，影响有效带宽占用率。

### 三、方案特色和创新（800 字以内）

#### 1、分级的虚拟机流量业务 QoS 保障解决方案

传统网络流量 QoS 保障通常是基于业务类型的方式，这是一种粗粒度的实现，在同一业务占满物理带宽的情况下很难区分相同业务的优先级。本方案将虚拟机业务流量服务质量分两级保障，分别为基于业务的带宽保障和基于虚拟机的带宽保障，其在为云中虚拟机业务流量提供自动化 QoS 保障的同时，尽可能提高物理链路利用率，有效解决了满带宽时同类业务 QoS 保障的问题，在更精细的层面上实现了对虚拟机业务流量的 QoS 保障。

#### 2、个性化流表自动下发机制

传统网络流量 QoS 保障实现中，SLA 通常被用来作为 QoS 输入的参数，最终生成的网络层策略往往是预先定义的、静态的。本方案在控制器层面上实现动态流表自动化下发机制，捕获 PACTET-IN 消息，根据当前网络拓扑中实时业务流量的情形，动态生成个性化流表并下发至交换机。

### 3、基于实时流量的端口队列带宽动态调整算法

在做基于业务的带宽保障时，动态调整实时流量的端口队列带宽，提出一个简单带宽协商自适应算法，有效解决队列带宽竞争时业务流量 QoS 保障问题。

### 4、基于流的细粒度 QoS 保障策略

在做基于虚拟机的带宽保障时，根据 openflow1.3 的新特性，实现对流的细粒度限速，设计一种基于权重的带宽分配策略，这里的权重影响因子是虚拟机优先级。

### 5、QoS 保障算法、策略动态管理

本方案系统框架采取模块化和热插拔思想，支持 QoS 保障策略和算法动态加载、更新和删除，系统管理人员能够自定义多种算法或者策略，实现策略算法模块和系统本身的高内聚与低耦合。

### 6、虚拟机业务流量实时通知

云环境下，网络带宽占用率成为影响数据密集型应用性能的关键因素。云租户需要知道当前虚拟机的网络流量状态以评估应用的性能。本系统设计网络流量通知模块，通知云租户虚拟机的网络流量状态。

## 四、应用具体设计论述（包括背景介绍、研究问题陈述、具体解决方法、设计架构图、预期实现目标等）

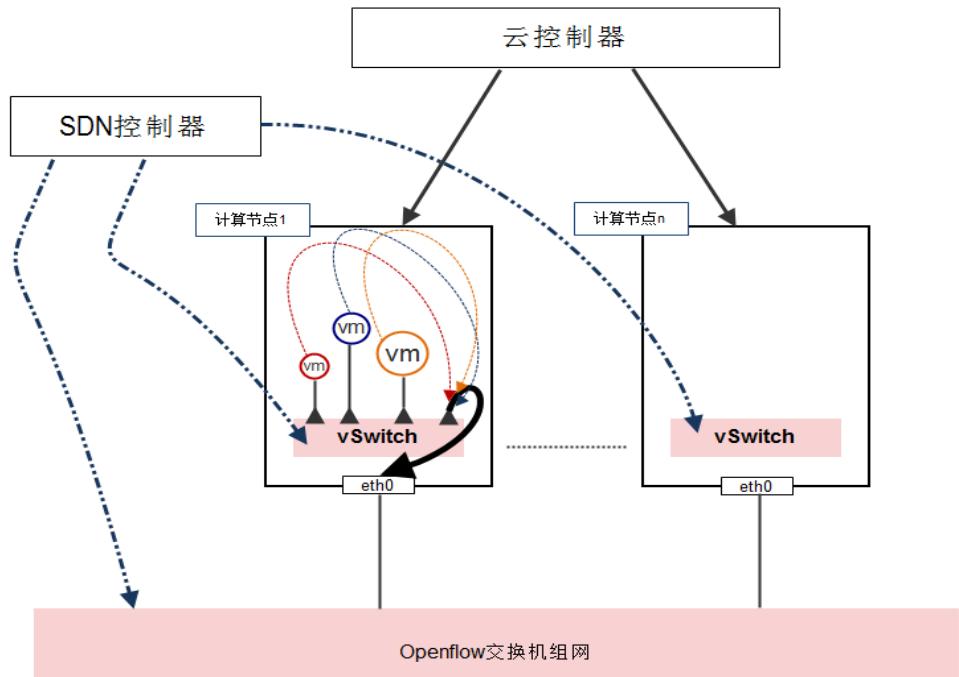
### 1. 背景介绍

动态性是云计算的最大特征之一，虚拟机可以随时启动、关闭、迁移，当前基于传统网络架构的模式在支撑云计算时缺乏一定的灵活性，尤其体现在云中业务流量 QoS 保障方面。云环境下，同一台宿主机上的多台虚拟机共享其物理网络接口，在数据密集型业务负载过重时，势必会引发对网络资源的竞争。传统 QoS 保障技术因敏捷和灵活性的天然缺失导致其无法很好地解决云中存在的问题，SDN 技术实现了控制平面与转发平面的分离，摆脱了硬件对网络架构的限制，实现网络层次较好的灵活性、敏捷性。除此之外，软件定义网络的可编程性带来了前所未有的灵活性，本应用借助 SDN，很好地解决了云环境下基于虚拟机业务流量的 QoS 保障问题。

### 2. 研究问题陈述

云计算基础设施层服务采取集中式控制的架构，由云控制节点管理多个计算节

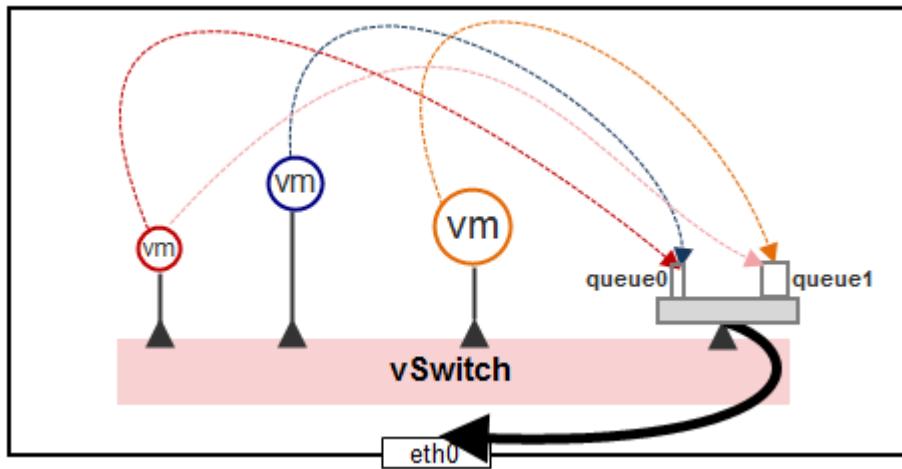
点，虚拟机运行在计算节点之上，云控制节点负责虚拟机具体调度，如图 4-1 所示。计算节点 1 上的多台虚拟机对宿主机的 eth0 网卡存在竞争，每台宿主机的 eth0 网卡直连在物理交换机上。本应用主要针对宿主机内部多台虚拟机流量做 QoS 保障，从而消除网络资源竞争带给虚拟机业务的影响。



(4-1 云计算架构)

### 3. 具体解决方法

为体现虚拟机流量业务等级，本方案对虚拟机业务进行分类，大致分为重要和非重要两类。为体现不同业务流量的优先级，本方案在宿主机物理网卡对应虚拟交换机的端口上做多个限速队列，共享物理带宽，如图 4-2 所示，即 eth0 对应 vSwitch 的端口。通过队列的带宽的大小实现不同虚拟机业务种类优先级的区分。业务流量通过 SDN 控制器将形成个性化流表，决策业务流匹配至不同队列，并下发至交换机。



(4-2 端口队列流量区分)

本方案在为虚拟机业务流量提供 QoS 保障的同时，尽可能提高链路利用率。故需满足以下情况：

- ① 优先保证重要业务的带宽，即不同种类业务竞争网络资源时，重要业务将尽可能占满带宽，非重要业务不参与竞争。
- ② 系统当前出现重要业务流量低谷时，非重要业务如若需要更多带宽，系统将动态对队列带宽动态调整，以满足非重要业务带宽需求。
- ③ 重要业务占满物理带宽时，重要业务之间也会出现网络资源竞争的情况，按虚拟机所属用户的类别保障其业务 QoS，即重要用户的虚拟机的重要业务将获得更多带宽资源。

因此，本应用提出基于分级的虚拟机流量业务 QoS 保障解决方案，两级保障如下：

#### ① 基于业务的带宽保障

对端口设置队列，在队列上做带宽限制，将业务分类并映射到队列中，队列具备优先级，重要的应用映射到优先级最高的队列中，优先级最高的队列中的服务将优先被保障，也就是说，优先级最高的队列在其需要最大带宽时将获得默认最大带宽的权利。为提供业务的带宽保障，本方案设计了基于实时流量的端口队列带宽动态调整算法。

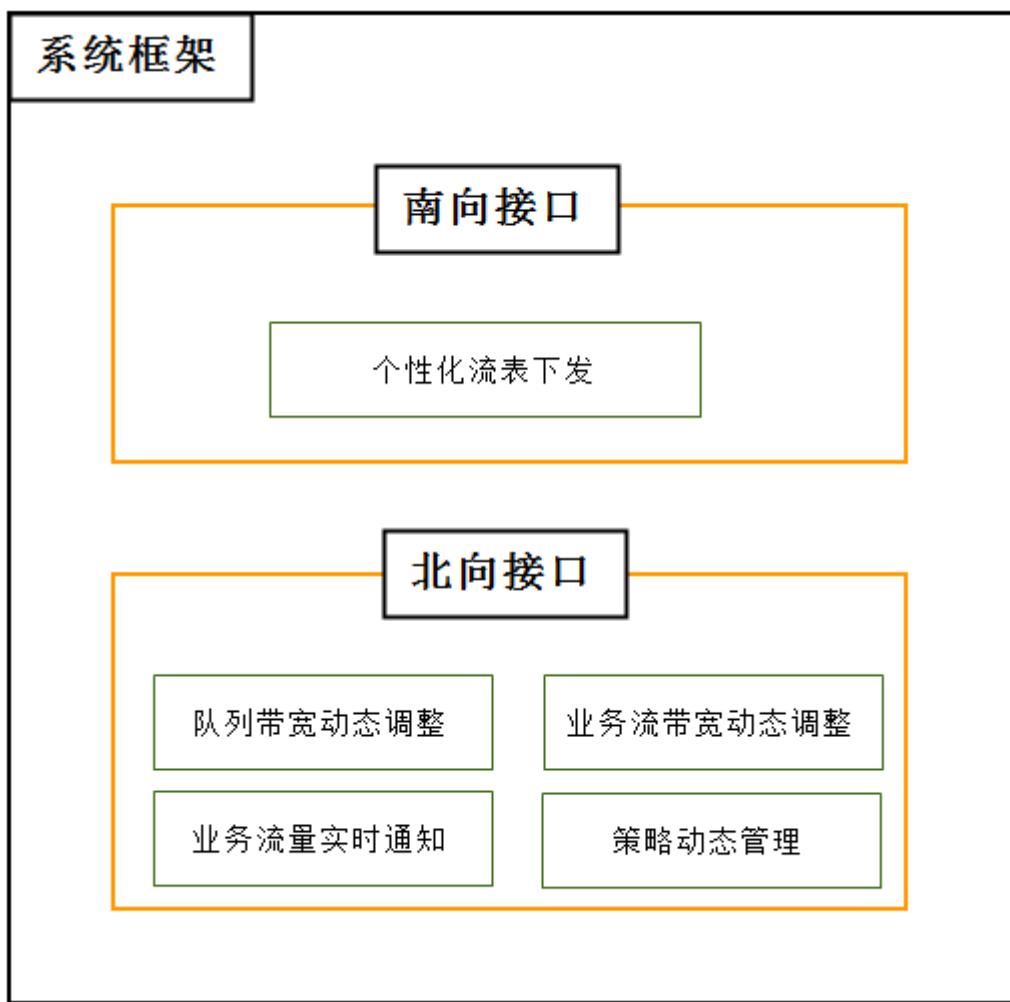
#### ② 基于虚拟机的带宽保障

当端口上的所有队列带宽比趋于稳态，并且每个队列带宽占用率处于饱和时，在相同队列里按不同虚拟机业务流的优先级做 QoS 保障，重要用户的虚拟机将拥有

更多的带宽。为提供虚拟机的带宽保障，本方案设计了基于流的细粒度 QoS 保障策略。

#### 4. 系统设计

本系统包括五大功能模块，分别为个性化流表下发模块、队列带宽动态调整模块、业务流带宽动态调整模块、业务流量实时通知模块以及策略动态管理模块，如图 4-3 所示。

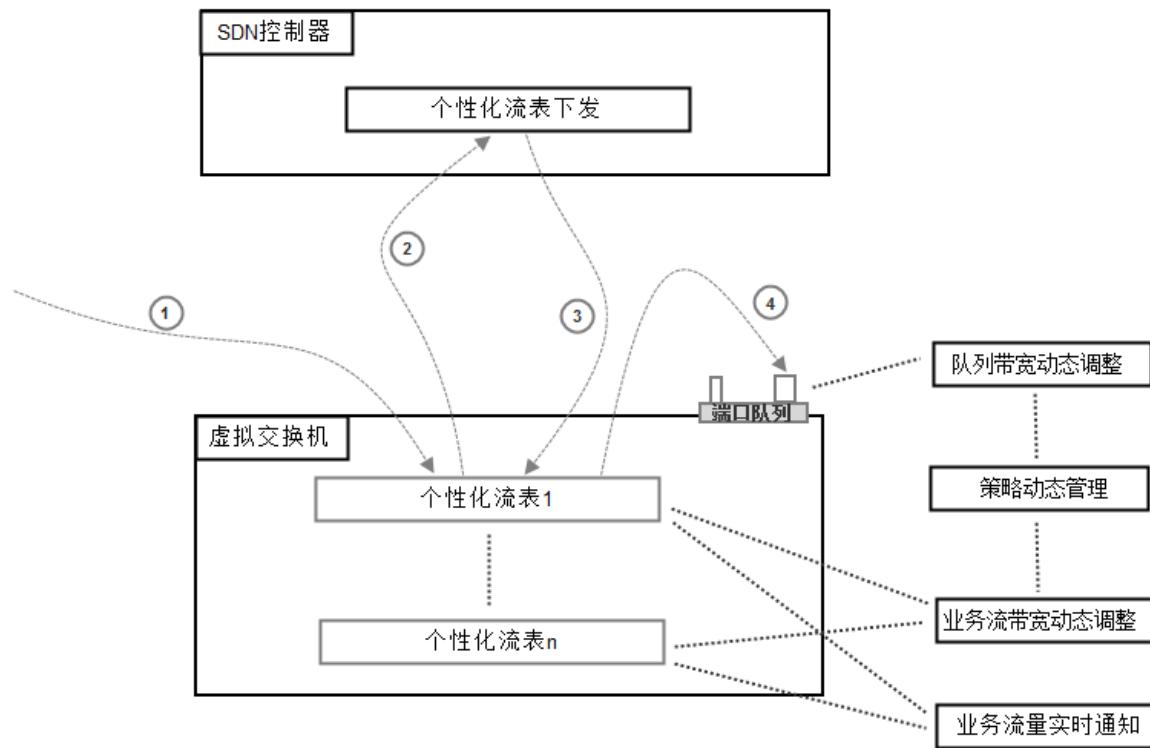


(图 4-3 系统框架图)

本系统工作流程如图 4-4 所示：

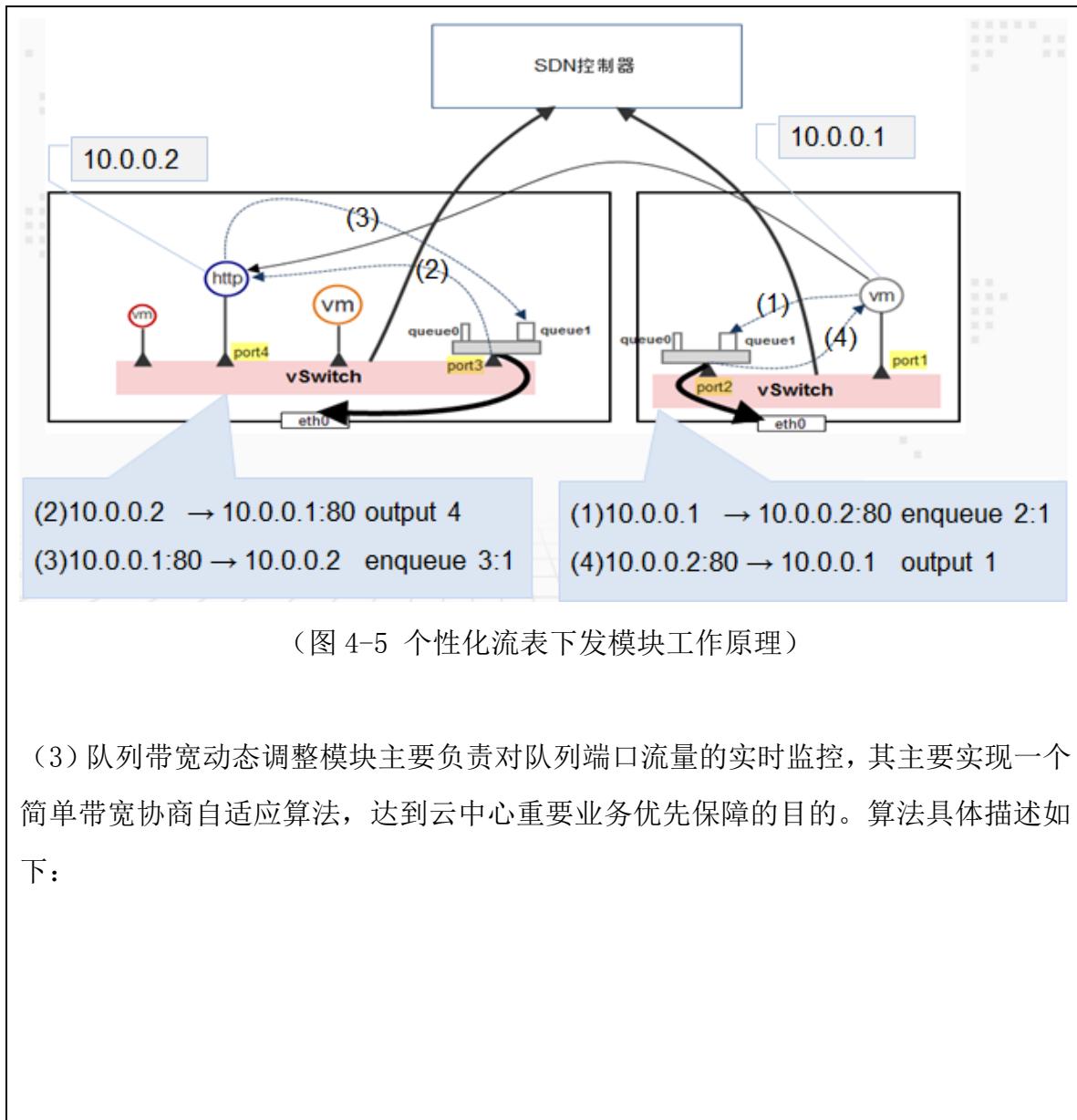
- (1) 虚拟机流量数据包发送至虚拟交换机，若无流表匹配，则发至控制器。
- (2) 控制器截获 Packet-in 消息，根据特征库分析其业务类型，决定转发至不同优先级的端口队列，监控端口队列流量情况，综合虚拟机类别，决策是否对业务流限速，同时生成个性化流表下发至虚拟交换机。这一过程由个性化流表下发模块完成，

基于 SDN 的南向接口实现。



(图 4-4 系统工作流程)

个性化流表下发模块接收到从 openflow 交换机发送过来的数据包，根据数据包分析其业务类型，进而调度到不同的队列，从而实现粗粒度的 QoS 保障，最大程度提高网络资源利用率，其主要工作原理如图 4-5 所示。当 IP 地址为 10.0.0.1 的虚拟机向位于不同计算节点的 IP 地址为 10.0.0.2 的虚拟机发送 Http 请求时，此时下发模块会解析数据包，判断其业务类型，从而在转发路径的节点下发流表，规划流量的转发细节。



(3) 队列带宽动态调整模块主要负责对队列端口流量的实时监控，其主要实现一个简单带宽协商自适应算法，达到云中心重要业务优先保障的目的。算法具体描述如下：

---

Algorithm1:<sup>④</sup>

---

```
Initialize nomalQueueBw, vipQueueBw, totalBw, factor +  
    satisfying nomalQueueBw + vipQueueBw = totalBw +  
While true do:  
    vipQueueTraffic <= read vip queue traffic +  
    normalQueueTraffic <= read normal queue traffic +  
    if  vipQueueTraffic == vipQueueBw then +  
        vipQueueBwAugment <= vipQueueBw * factor +  
        vipQueueBw <= vipQueueBwAugment < totalBw ? vipQueueBwAugment : totalBw +  
        nomalQueueBw <= totalBw - vipQueueBw +  
        set normal queue bandwidth = nomalQueueBw +  
        set vip queue bandwidth = vipQueueBw +  
    else +  
        if  normalQueueTraffic == nomalQueueBw then +  
            nomalQueueBwAugment <= nomalQueueBw * factor +  
            if  (totalBw - nomalQueueBwAugment) > vipQueueTraffic then +  
                nomalQueueBw <= nomalQueueBwAugment +  
                vipQueueBw <= totalBw - nomalQueueBw +  
                set normal queue bandwidth = nomalQueueBw +  
                set vip queue bandwidth = vipQueueBw +  
            endif +  
        endif +  
    endif +  
    sleep 5 seconds +  
endwhile
```

---

(图 4-6 队列带宽动态调整算法)

(4) 业务流带宽动态调整模块主要解决当重要端口队列占满物理带宽时, 根据用户虚拟机的优先级对若干业务流进行限速, 重要级别低的若干虚拟机业务流将被限速。具体算法如下:

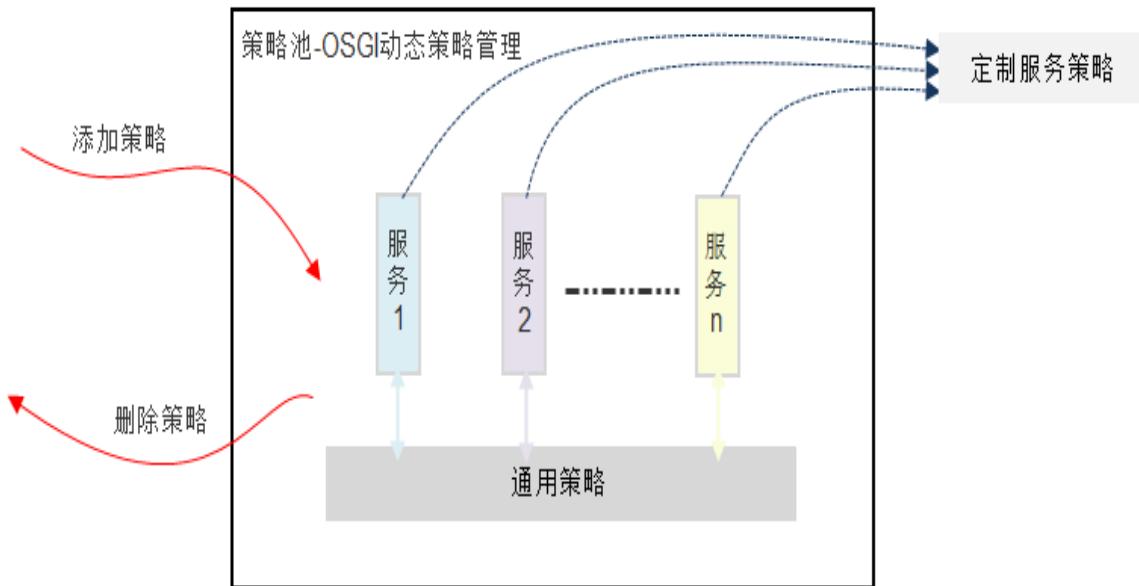
```

Algorithm2
Initialize vipQueueBw, totalBw, vmNumFactor, trafficFactor
Initialize vmPriorityQueue denoting VM priority queue
While true do
    vipQueueTraffic <- read vip queue traffic
    if  vipQueueTraffic == vipQueueBw then
        vmPriorityQueue <- get all vms belong to vip queue
        limitedVmsNumber <- vmPriorityQueue.length * vmNumFactor
        limitedVms[] <- get the lowest limitedVmsNumber Vms from vmPriorityQueue
        for i in limitedVms
            currentVmTraffic <- read current traffic from flow table
            currentVmTrafficLimit <- currentVmTraffic * trafficFactor
            do traffic limiting
        endfor
    endif
    sleep 5 seconds
endwhile

```

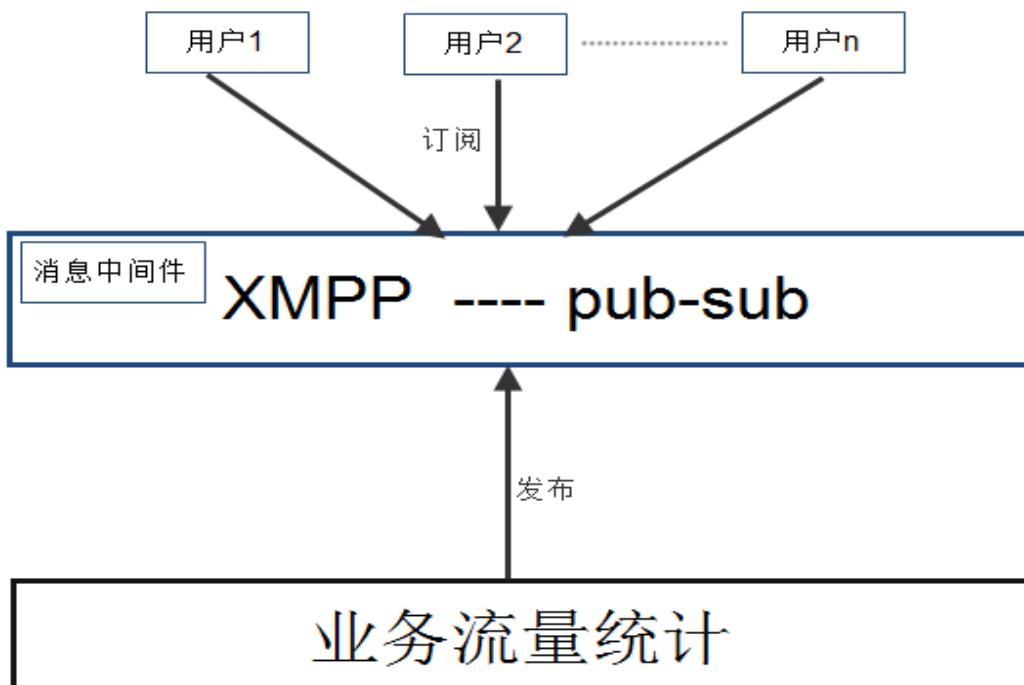
(图 4-7 业务流带宽动态调整算法)

(5) 动态策略管理模块主要负责维护多种 QoS 策略的动态加载和删除。网络管理人员可以根据业务的变化随时添加和删除自定义的 QoS 策略, 即时生效。同一服务在不同计算节点不同用户的 SLA 需求不同, 同一服务在不同计算节点可能同一用户的 SLA 需求也不一样, 导致同一服务在不同节点的 QoS 保障规则不同, 如 MapReduce 任务, 不同节点之间通信量大小不同, 数据通信量大的节点需要高带宽。当重要业务占满带宽时, 业务流带宽动态调整模块中基于流的带宽分配考虑业务本身的影响因子, 该因子的生成根据定制服务策略产生。本模块基于 OSGI 框架实现, 如图 4-8 所示。



(图 4-8 动态策略管理)

(6) 本系统还提供业务流量实时通知功能, 帮助运行数据密集型应用的用户对应用本身做性能分析。本模块基于消息中间件的 pub-sub 模式实现。



(图 4-9 业务流量实时通知模块)

## 五、应用实现过程（包括程序流程、实验设计以及实验过程和结果）

本实验旨在验证云环境下同一台宿主机上的多台虚拟机对物理网络资源存在竞争时 Qos 策略对虚拟机业务流量保障的有效性和正确性。

### 1. 实验环境

硬件环境：

共四台服务器节点，分别为两台计算节点、一台控制节点以及一台网络节点，其硬件配置均为：8 核 CPU、32G 内存、120G 磁盘空间大小。

软件环境：

- (1) 云环境：openstack havana 版本
- (2) SDN 控制器：floodlight
- (3) 虚拟交换机：openvSwitch

在云环境中启动八台虚拟机，分别接入在两台交换机下，各 4 台虚拟机。虚拟交换机使用 openvswitch 实现，由 floodlight 开源控制器控制。虚拟机操作系统为 Ubuntu server 14.04，软件环境如下：

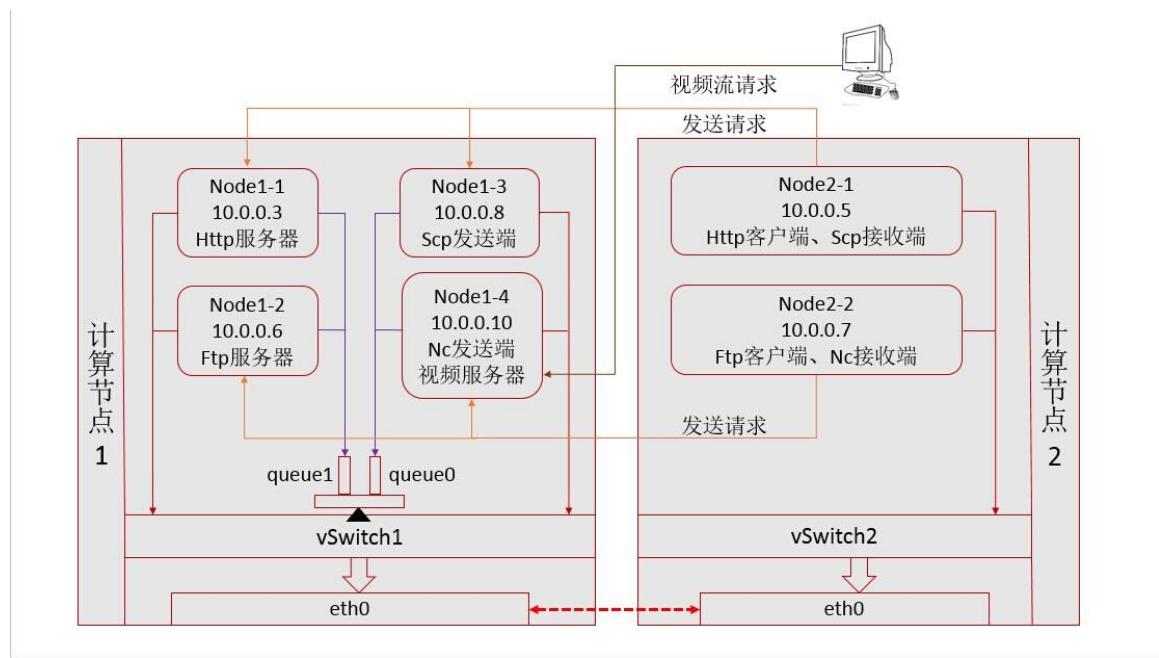
表 5-1 实验环境

OpenvSwitch1			OpenvSwitch2		
Hostname	Ip	服务	Hostname	Ip	服务
node1-1	10.0.0.3	Http 服务器	node2-1	10.0.0.5	Http 客户端 Scp 接收端
node1-2	10.0.0.6	Ftp 服务器	node2-2	10.0.0.7	Ftp 客户端 Nc 接收端
node1-3	10.0.0.8	Scp 发送端			
node1-4	10.0.0.10	Nc 发送端 视频流服务器	External-node		视频流客户端

## 2. 实验设计

本实验基于 OpenStack 搭建云计算基础设施环境，共两个计算节点，在计算节点 1 上启动四台虚拟机分别安装不同类型业务，提供不同服务，在计算节点 2 上启动两台虚拟机作为访问计算节点 1 上虚拟机业务服务的客户端，按表 5-1 部署实验环境如图 5-2 所示。

在计算节点 1 虚拟交换机 openvswitch1 对应物理网卡的端口上做两个队列 queue0 和 queue1，queue0 承载非重要业务，queue1 承载重要业务。针对虚拟机业务分类，计算节点 1 上 Http 服务和 Ftp 服务为重要业务，分别占用端口号 80 和 21，根据个性化流表下发模块工作方式，这些流量流经 queue1。其上的 Scp 发送端、Nc 发送端以及视频流服务被认为是非重要业务，流量经过 queue0。



(图 5-2 虚拟机业务部署)

在计算节点 2 的虚拟机中分别请求计算节点 1 中的虚拟机业务，不同宿主机上的虚拟机间通信需要经过物理网卡，即 queue0 或者 queue1，模拟虚拟机业务流量在队列中的竞争情况，根据队列带宽动态调整模块，及时调整 queue0 和 queue1 的带宽比例，达到重要业务优先保障的目的。

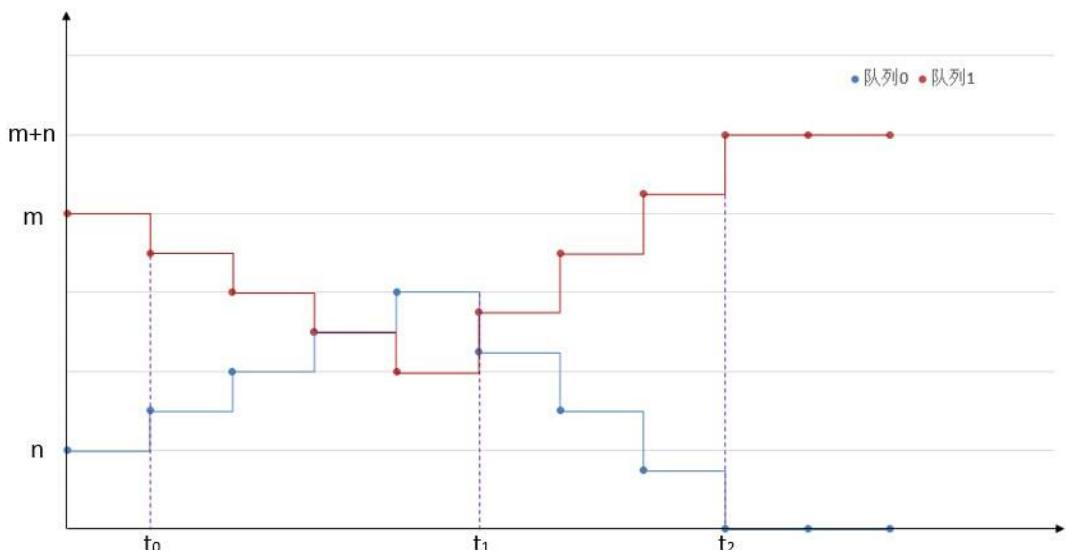
本实验将验证以下虚拟机业务流量保障情景：

- ① 当 queue0 中业务流量占满带宽且 queue1 空闲时，队列带宽动态调整策略将向上

调整 queue0 和 queue1 带宽比例。

② 当 queue0 中含有业务流量且 queue1 业务流量增长至其带宽限制时, 队列带宽动态调整策略将通过向下调整 queue0 和 queue1 带宽比例, 从而优先保证 queue1 中虚拟机业务流量。

本实验期望带宽调整过程如图 5-3 所示, queue0 初始带宽  $n$ , queue1 初始带宽  $m$ 。在  $t_0$  时刻, 模拟 queue0 业务流量, queue0 带宽占满且 queue1 空间, 队列带宽动态调整策略将向上调整 queue0 和 queue1 带宽比例。在  $t_1$  时刻, 模拟 queue1 业务流量, queue1 带宽迅速占满, 然而 queue0 依然保持业务流量, 队列带宽动态调整策略将优先保证 queue1 中虚拟机业务流量, 向下调整 queue0 和 queue1 带宽比例。



(图 5-3 队列带宽调整过程)

### 3. 实验过程及结果

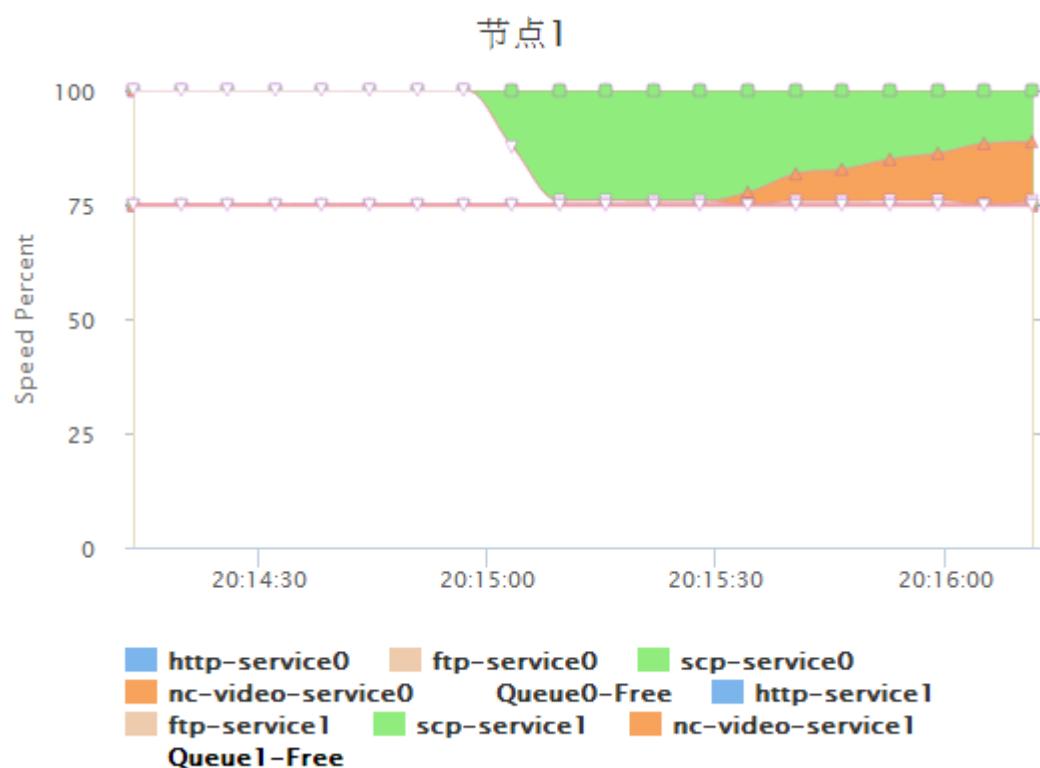
① 在云环境中网络资源充裕条件下, 测量单客户端请求每种类型虚拟机业务服务时产生流量所占带宽大小, 其结果如下:

服务	平均传输速率
Http 服务	32MB/s
Scp 服务	25MB/s

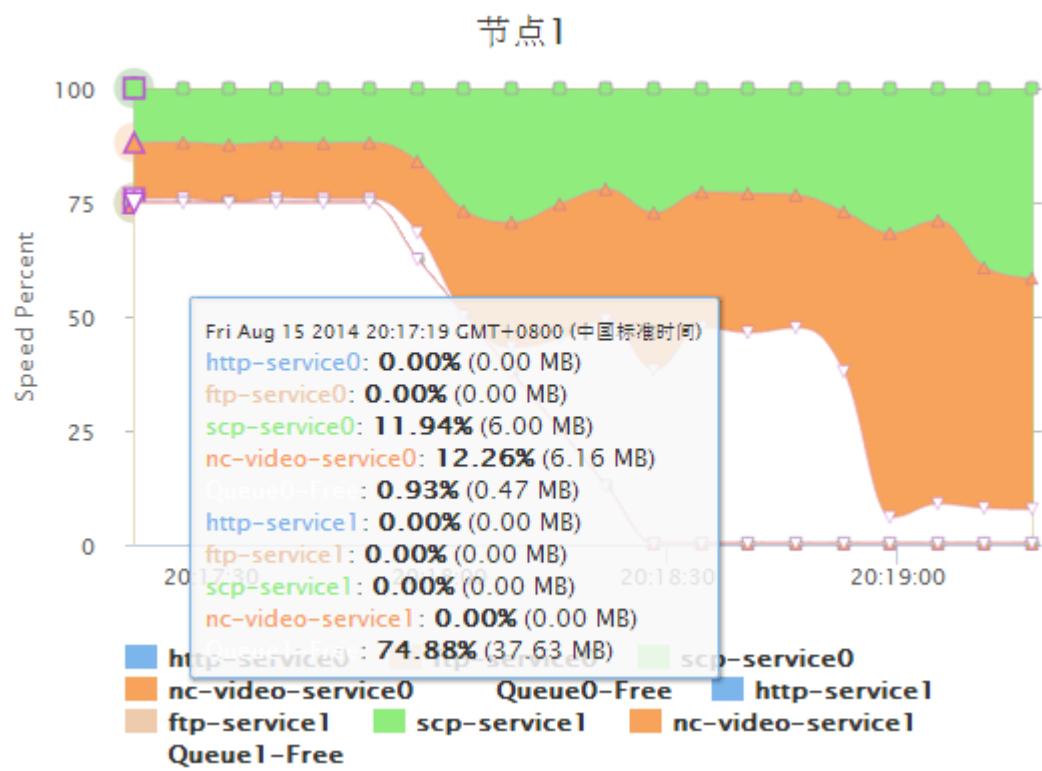
Ftp 服务	40MB/s
Nc 服务	40MB/s
视频流服务	1MB/s

② 为简化实验, 尽可能使用较少的服务占满队列带宽, 设置云环境中每台计算节点上虚拟交换机上物理网卡对应的端口队列 queue0 大小为 100Mb/s, queue1 大小为 300Mb/s。在同一时间请求 Scp 和 Nc 服务将占满所有队列带宽总和。同样, 同一时间请求 Http 和 Ftp 服务也将占满所有队列带宽总和。

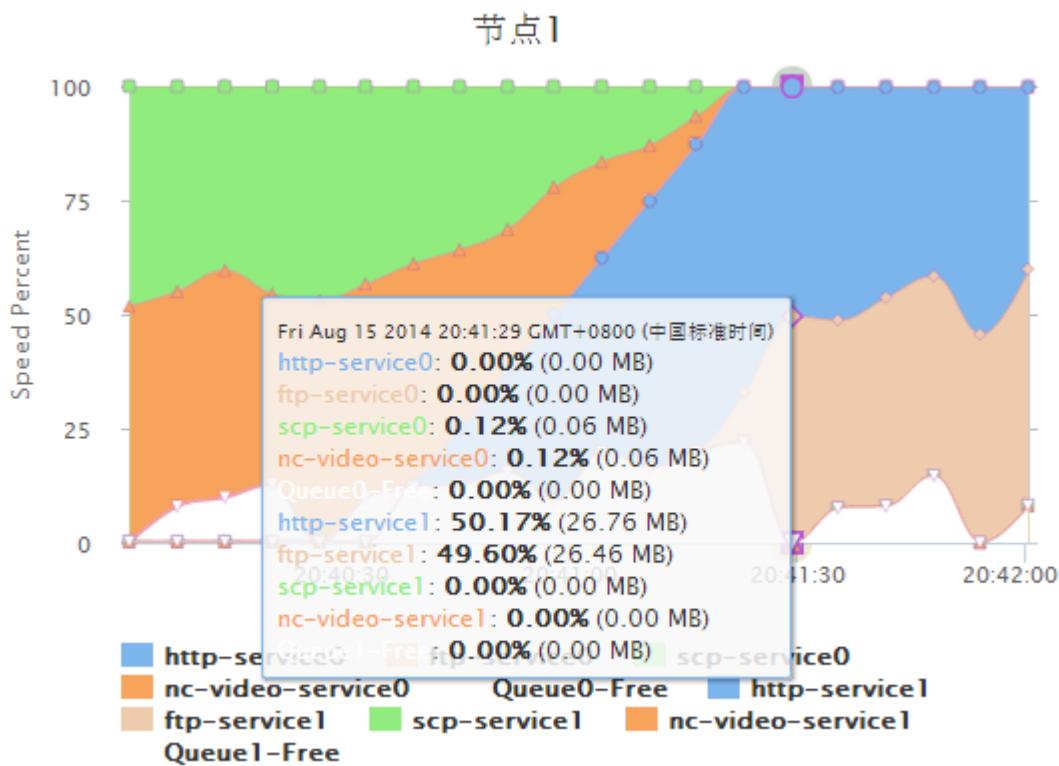
③ 在计算节点 2 的虚拟机 node2-1 上请求 Scp 服务, 其流量经过 queue0, 受 queue0 带宽限制, 流量大小约为 12MB/s。在计算节点 2 的虚拟机 node2-1 上请求 Nc 服务, Nc 业务流量将与 Scp 业务流量在 queue0 中形成带宽竞争。与此同时, 广域网下访问计算节点 1 虚拟机 node1-4 的视频流服务。



④ 开启队列带宽动态调整模块, 因 queue1 中并无流量产生, 策略将增长 queue0 和 queue1 的比例, 直至占满物理带宽。



- ⑤ 在计算节点 2 的虚拟机 node2-1 上请求 Http 服务，其流量经过 queue1，因 Scp 业务和 Nc 业务占据大部分带宽，此时 queue1 带宽很小，策略将向下调整 queue0 和 queue1 的比例，与此同时，在 node2-2 上访问 Ftp 服务，Http 和 Ftp 服务将占满整个物理带宽，即 queue1 占满整个物理带宽。视频流服务、Scp 服务以及 Nc 服务的流量将被限制。



Http 服务和 Ftp 服务流表自动下发情况如下：

在计算节点 1 上虚拟交换机中流表

Floodlight								Live updates			
				Dashboard		Topology		Switches		Hosts	
9007199254740992	0	port=83, VLAN=-1, src=fa:16:3e:59:0f:b1, dest=fa:16:3e:76:a7:e6		output	0	0	3 s	5 s			
0	-1	port=83, VLAN=-1, src=fa:16:3e:bd:ee:55, dest=fa:16:3e:fd:52:00, ethertype=0x0800, proto=6, IP src port=-15770, IP dest port=20, src=10.0.0.7, dest=10.0.0.6, TOS=2		output	0	0	51 s	1800 s			
0	-1	port=109, VLAN=-1, src=fa:16:3e:fd:52:00, dest=fa:16:3e:bd:ee:55, ethertype=0x0800, proto=6, IP src port=20, IP dest port=-15770, src=10.0.0.6, dest=10.0.0.7, TOS=2		enqueue	0	0	51 s	1800 s			
0	-1	port=83, VLAN=-1, src=fa:16:3e:fb:55:71, dest=fa:16:3e:ff:5c:ce, ethertype=0x0800, proto=6, IP src port=-18536, IP dest port=80, src=10.0.0.5, dest=10.0.0.3, TOS=0		output	147141	9770746	1406 s	1800 s			
0	-1	port=108, VLAN=-1, src=fa:16:3e:ff:5c:ce, dest=fa:16:3e:fb:55:71, ethertype=0x0800, proto=6, IP src port=80, IP dest port=-18536, src=10.0.0.3, dest=10.0.0.5, TOS=0		enqueue	649647	983563441	1406 s	1800 s			
0	-1	port=83, VLAN=-1, src=fa:16:3e:bd:ee:55, dest=fa:16:3e:fd:52:00, ethertype=0x0800, proto=6, IP src port=-28147, IP dest port=21, src=10.0.0.7, dest=10.0.0.6, TOS=4		output	0	0	78 s	1800 s			
0	-1	port=109, VLAN=-1, src=fa:16:3e:fd:52:00, dest=fa:16:3e:bd:ee:55, ethertype=0x0800, proto=6, IP src port=21, IP dest port=-28147, src=10.0.0.6, dest=10.0.0.7, TOS=4		enqueue	0	0	78 s	1800 s			

在计算节点 2 上虚拟交换机中流表

Floodlight Dashboard Topology Switches Hosts  Live updates

Cookie	Priority	Match	Action	Packets	Bytes	Age	Timeout
0	-1	port=114, VLAN=-1, src=fa:16:3e:fd:52:00, dest=fa:16:3e:bd:ee:55, ethertype=0x0800, proto=6, IP src port=21, IP dest port=-28147, src=10.0.0.6, dest=10.0.0.7, TOS=4	output 143	0	0	28 s	1800 s
0	-1	port=114, VLAN=-1, src=fa:16:3e:ff:5c:ce, dest=fa:16:3e:fb:55:71, ethertype=0x0800, proto=6, IP src port=80, IP dest port=-18536, src=10.0.0.3, dest=10.0.0.5, TOS=0	output 142	532987	975856641	1357 s	1800 s
0	-1	port=114, VLAN=-1, src=fa:16:3e:fd:52:00, dest=fa:16:3e:bd:ee:55, ethertype=0x0800, proto=6, IP src port=20, IP dest port=-15770, src=10.0.0.6, dest=10.0.0.7, TOS=2	output 143	0	0	1 s	1800 s
0	-1	port=142, VLAN=-1, src=fa:16:3e:fb:55:71, dest=fa:16:3e:ff:5c:ce, ethertype=0x0800, proto=6, IP src port=-18536, IP dest port=80, src=10.0.0.5, dest=10.0.0.3, TOS=0	enqueue 114:1	147141	9770746	1357 s	1800 s
0	-1	port=143, VLAN=-1, src=fa:16:3e:bd:ee:55, dest=fa:16:3e:fd:52:00, ethertype=0x0800, proto=6, IP src port=-15770, IP dest port=20, src=10.0.0.7, dest=10.0.0.6, TOS=2	enqueue 114:1	0	0	1 s	1800 s
0	-1	port=143, VLAN=-1, src=fa:16:3e:bd:ee:55, dest=fa:16:3e:fd:52:00, ethertype=0x0800, proto=6, IP src port=-28147, IP dest port=21, src=10.0.0.7, dest=10.0.0.6, TOS=4	enqueue 114:1	0	0	28 s	1800 s

### 页面效果展示：



### 队列带宽信息-Log

```
2014-08-15 20:44:54 当前 节点1交换机下的带宽 q1=43.875MB q0=6.375MB 节点2交换机下的带宽 q1=37.625MB q0=12.625MB
2014-08-15 20:45:00 当前 节点1交换机下的带宽 q1=43.875MB q0=6.375MB 节点2交换机下的带宽 q1=37.625MB q0=12.625MB
2014-08-15 20:45:07 当前 节点1交换机下的带宽 q1=43.875MB q0=6.375MB 节点2交换机下的带宽 q1=37.625MB q0=12.625MB
2014-08-15 20:45:13 当前 节点1交换机下的带宽 q1=43.875MB q0=6.375MB 节点2交换机下的带宽 q1=37.625MB q0=12.625MB
2014-08-15 20:45:19 当前 节点1交换机下的带宽 q1=43.875MB q0=6.375MB 节点2交换机下的带宽 q1=37.625MB q0=12.625MB
2014-08-15 20:45:25 当前 节点1交换机下的带宽 q1=43.875MB q0=6.375MB 节点2交换机下的带宽 q1=37.625MB q0=12.625MB
2014-08-15 20:45:32 当前 节点1交换机下的带宽 q1=43.875MB q0=6.375MB 节点2交换机下的带宽 q1=37.625MB q0=12.625MB
2014-08-15 20:45:38 当前 节点1交换机下的带宽 q1=43.875MB q0=6.375MB 节点2交换机下的带宽 q1=37.625MB q0=12.625MB
```

### 流表下发情况-Log

```
2014-08-15 20:43:00 00:00:00:e0:81:b7:be:73 [ SourceMac fa:16:3e:fd:52:0 DestinationMac fa:16:3e:bd:ee:55 SourceIp 10.0.0.6 DestinationIp 10.0.0.7
SourcePort 20 DestinationPort -14077 Action:output ]
2014-08-15 20:43:00 00:00:00:e0:81:b7:bf:3f [ SourceMac fa:16:3e:fd:52:0 DestinationMac fa:16:3e:bd:ee:55 SourceIp 10.0.0.6 DestinationIp 10.0.0.7
SourcePort 20 DestinationPort -14077 Action:enqueue1 ]
2014-08-15 20:43:00 00:00:00:e0:81:b7:bf:3f [ SourceMac fa:16:3e:bd:ee:55 DestinationMac fa:16:3e:fd:52:0 SourceIp 10.0.0.7 DestinationIp 10.0.0.6
```

### 六、附录部分（相关实验图表和数据、源代码、演示视频等演示性和验证性材料） (此处给出附件清单列表即可)

Qos 策略源代码: SdnTrafficQos.rar

云中流量演示源代码: TestCloudTraffic.rar

演示视频: SDN-video.wmv

