2012 13th International Conference on Parallel and Distributed Computing, Applications and Technologies

Application Utility-based Bandwidth Allocation Scheme for Data Center Networks

Yangyang Li*†, Hongbo Wang*†, Jiankang Dong*, Shiduan Cheng*
*State Key Laboratory of Networking and Switching Technology
Beijing University of Posts and Telecommunications, Beijing, 100876, China
†State Key Laboratory of Software Development Environment
BeiHang University, Beijing, 100191, China
Email: {yyli, hbwang, dongjk, chsd}@bupt.edu.cn

***Abstract*—The key infrastructure of Cloud Computing is data center which is shared by many tenants. Each tenant's application competes for acquiring more network bandwidth in order to maximize its utility. However, this may cause interference among these diverse applications. Malicious competition not only degrades its performance, but also makes the overall performance of the data center poor and ineffective. To ensure the Quality of Services (QoS) and achieve high network utilization, in this paper, we propose a bandwidth allocation scheme for data center networks (DCNs), which is based on an application utility-based model. In our scheme, multi-path feature of DCN is leveraged to improve the network utilization, and utility functions are constructed to differentiate the throughput and delay sensibilities of different applications. Moreover, our scheme is suitable for arbitrary DCN topologies and without modification on current hardware. The numerical simulation shows that our scheme can provide bandwidth guarantee, fine-grained service differentiation and achieve high network utilization.**

***Keywords*-Cloud Computing, Data Center Networks, Application Utility, Bandwidth Allocation**

I. Introduction

In recent years, Cloud Computing has been considered as the revolutionary technology of IT industry by the public. According to the definition of UC Berkeley [1], Cloud Computing refers to the applications delivered as pay-as-you-go services over the Internet, as well as the hardware and system software in the data centers that provide those services. Among these, the key infrastructure of Cloud Computing is data center which contains a large number of computers interconnected by a data center network, related hardware, and system softwares.

In terms of providing cost-effective pay-as-you-go services, Cloud Computing is ambitious, because it is in accordance with the economic law: the economy of scale can be obtained through large-scale purchasing, centralized operation management and statistical multiplexing of resources (the cost decreases to $1/5 - 1/7$ the prices offered to a medium-sized data center [1]). From the viewpoint of who owns the data centers, it is cost-effective to maximize the resource utilization on the basis of offering on-demand using of resources for Cloud Computing.

Currently, in order to improve the resource utilization and reduce the cost of management, cloud data centers mostly offer on-demanding computing and storage resources by utilizing virtualization technology [2] to consolidate servers within the data center. However, the use and management of bandwidth in DCN has not been fully resolved. Recent measurement and research [3] [4] indicate that, bandwidth is becoming an important factor which affects the performance of data center. On one hand, much idle bandwidth has not been effectively used, on the other hand, the Quality of Services has not been guaranteed. How to manage and especially allocate the bandwidth within data centers is becoming one of the most important issues which need to be addressed urgently in the research field of Cloud Computing.

The design of a DCN bandwidth allocation scheme should take the benefits of both Cloud Providers (CPs) and tenants into consideration: from the perspective of CPs, the bandwidth should be utilized as much as possible to maximize their revenue which come from their investment in networking equipment; while from the viewpoint of tenants, the Service Level Agreement (SLA) should be met to guarantee the quality of their services.

Many bandwidth allocation schemes [5] [6] [7] [8] [9] have been proposed, nevertheless, some of them [5] [6] cannot improve the benefit of CPs as the bandwidth in their schemes is not sufficiently used and the others [7] [8] [9] cannot offer differentiated QoS according to the application types of tenants, as a result, their service differentiation is coarse-grained. A novel DCN bandwidth allocation scheme should be designed to meet the practical requirements.

We suppose that the scheme of bandwidth allocation for DCNs should achieve the following goals:

1) *High utilization:* to improve the utilization of networking resources, idle bandwidth must be sufficiently utilized.

2) *Fine-grained service differentiation:* to meet different QoS requirements, fine-grained differentiated bandwidth allocation should be offered according to the application types of different tenants.

3) *Easy deployment:* to deploy the scheme, the hardware of servers and networking equipment within data centers should not be changed.

In addition, the following features of cloud data center open the design space for bandwidth allocation:

1) *Multi-Path:* in the past years, in order to improve the communication capacity between servers, researchers in [10]

978-0-7695-4879-1/12 $26.00 © 2012 IEEE
DOI 10.1109/PDCAT.2012.55

IEEE computer society

[11] [12] have proposed novel architectures for DCNs. To eliminate the oversubscription, all of them reduce the mismatch of lower and upper links by offering redundant links.

2) *Unified management entity:* as a cloud data center is operated and administrated by a single organizer, CPs usually use Centralized Management Units (CMU) to monitor the status of data center and implement related management operation.

3) *Edge switching:* with the development of virtualization technology, edge switching equipment is shifted from the access switch to physical server where the virtual machines are contained. The first hop forwarding equipment is replaced by the virtual machine monitor (e.g. Hypervisor).

In order to achieve above design goals, in this paper, we propose an application utility-based bandwidth allocation scheme for DCNs, combining with the characteristics of cloud data center: to improve the utilization of network, mapping the bandwidth allocated to different applications on multi-path; to provide fine-grained service differentiation, mapping the bandwidth allocated to different applications to utility functions by constructing utility functions based on the throughput-sensitive and delay-sensitive feature of applications; to keep deployable, using CMU to compute the result of the allocation, then informing the Hypervisor to deploy our scheme. Our scheme can run without the modification on current hardware. The numerical simulation indicates that our scheme satisfies the scheduled goals.

The remainder of this paper is organized as follows. After presenting the related work in Section II, we describe the architecture and system model of our scheme in Section III and Section IV. Then we give an example of our allocation scheme by numerical simulation in Section V. Finally, we conclude the paper in Section VI.

II. Related work

Recently, some progress [5] [6] [7] [8] [9] has been made in the field of bandwidth allocation and service differentiation for DCN. The authors in [5] offer three priorities bandwidth allocation including type 0, type 1 and best effort. Type 0 provides guaranteed bandwidth between two VMs, which is analogous to Integrated Service [13]. Type 1 provides only last and/or first hop guarantee, and the best effort without guarantee. The authors in [6] propose two classes virtual network abstractions that cater to application requirements. But they only distinguish data-sensitive applications from the others. Even for the highest priority applications, both of the methods allocate fixed bandwidth only. If there exists idle bandwidth, it will be used by best effort application. The bandwidth is far from fully utilized, and obviously, the priority they offer is too coarse-grained.

The authors in [9] and [7] allocate bandwidth according to different weights of applications in centralized and distributed way respectively. The weight based method cannot differentiate the types of different applications, meanwhile, the diversity of applications in DCN causes that there is not exist an effective standard to ensure weights for every application.

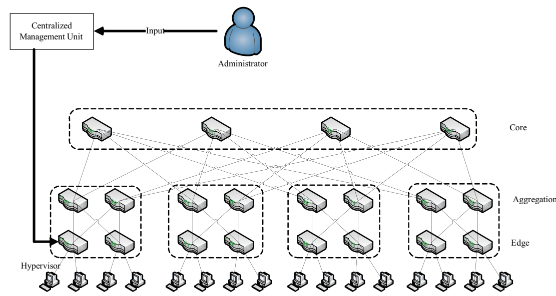

Fig. 1. A simple fat-tree topology

The authors in [8] summarize partial literature aforementioned, and they propose a bandwidth allocation scheme only uses physical servers. Their fundamental assumption is the links between all servers are un-blocking, and this corresponds less with reality which limits their usage.

In this paper, we firstly apply application utility-based model to design a bandwidth allocation scheme for DCN. We leverage the multi-path feature of DCN and make full use of idle bandwidth, compared to [5] [6], our scheme has higher bandwidth utilization; we construct utility functions to reflect the throughput-sensitive and delay-sensitive features of different applications, we believe that allocate bandwidth according to the feature of applications will give fine-grained service differentiation compared to [7] [9]; meanwhile, our scheme can be used without the limitation of [8].

III. Overview

A. Topology

To describe convenience in this paper, we take one of the classic DCN topology fat-tree [11] as an example to present our scheme. It should be noted that our scheme can be easily popularized to arbitrary DCN topologies. As shown in Fig. 1, the fat-tree is split into three layers, which is labeled edge/access, aggregation and core respectively. There are k pods, each containing two layers of $k/2$ switches. Each k-port switch in the lower layer is directly connected to $k/2$ servers. Each of the remaining $k/2$ ports is connected to $k/2$ of the k ports in the aggregation layer of the hierarchy. There are $(k/2)^2$ k-port switches. Each core switch has one port connect to each of k pod. In general, a fat-tree built with k-port switches supports $k^3/4$ servers. In Fig.1, $k = 4$, so it can support 16 servers.

B. Centralized Management Unit

CMU is the computing entity of the whole allocation scheme, whose main duty is to maintain the routing matrix (RM) and allocate the bandwidth according to the RM and application requirements input by the administrator. The application requirements should include the following parameters at least: throughput-sensitive parameter, delay-sensitive parameter and the lower bound of the bandwidth requirement. The specific meaning of these parameters we will introduce in the next section.

Because the routing matrix involves how to number every links in the topology, in this paper, we number the links with the method similar to [11], where they used to allocate IP addresses to switches. In general, begin with the pod 1, giving the links between the edge layer and the aggregation layer with the number $1 - k/2$ (from left to right). After this, number the links between the aggregation layer and the core layer in the same way. The remainder pods could be numbered in sequence. In Section V, we will give an example of a numbered topology.

CMU computes the solution of our application utility-based model, and then informs the Hypervisor to deploy. In Fig. 1, the hypervisor is denoted by the edge switch.

C. Hypervisor

Hypervisor, which is also named Virtual Machine Monitor, is one of the hardware virtualization techniques allowing multiple operating systems to run concurrently on a host computer. In a virtualized computing environment, hypervisors take over the duty to process packet first before the packet be sent out or received by virtual machine which is located in corresponding physical machine. In our scheme, it is hypervisor to actually execute bandwidth allocation according to the results informed by CMU.

Hypervisors distribute traffic to multiple paths by using rate-limiting and multi-path routing. We discuss the implementation after presenting our system model in the next Section.

IV. System model

We model the DCN topology as a weighted undirected graph and denote it as $G = (N, L)$, where N is the set of switches and L is the set of physical links, denoted by $L = 1, 2, ..., l(l \geq 2)$. Define the bandwidth capacity vector $C = (c_1, c_2, ..., c_l)(l \geq 2)$ and residual capacity vector $\gamma = (\gamma_1, \gamma_2, ..., \gamma_l)(l \geq 2)$, c_i and γ_i represent the bandwidth capacity and residual capacity of physical link i respectively. We use VM-VM pairs to represent the communication between each virtual machine. An application uses several virtual machine in common, which means using several VM-VM pairs. To simplify the problem, in this paper, we assume each application uses two VMs, and contains one VM-VM pairs. Even applications have multiple VMs, the utility can be calculated cumulatively. Hereafter, the term VM-VM pair and application will be used interchangeably. The set of VM-VM pairs is denoted by $I = 1, 2, ..., n(n \geq 2)$ corresponding to n applications respectively. Since in Fat-tree topology, each VM can communicate with each other using multiple paths. The quantity of the paths is determined by the quantity of core switches, we use k to represent it. So we can define bandwidth allocation vectors $X_i = (x_{i1}, x_{i2}, ..., x_{ik})(i \geq 2, k \geq 2)$, i represents the i_{st} VM-VM pair, x_{ij} represents the bandwidth allocated on path j for the VM-VM pair i, which also means the i_{th} application in the context. Then the bandwidth allocation matrix can be denoted by $X = (X_1, X_2, , X_n)(n \geq 2)$. In order to guarantee the minimum bandwidth requirements of applications, we define the lower bound of bandwidth allocation as vector $low = (low_1, low_2, ..., low_n)(n \geq 2)$, The bandwidth allocation should satisfy the following inequality:

$$\sum_{j=1}^{k} x_{ij} \geq low_i \quad for\ all\ i \in I \tag{1}$$

It means that the sum of bandwidth on k paths, i.e., the bandwidth allocated to the application has to meet the minimum requirement.

The routing matrix can be denoted by a link matrix:

$$R_{l,nk} = \begin{pmatrix} R_{1,1} & R_{1,2} & \dots & R_{1,nk} \\ R_{2,1} & R_{2,2} & \dots & R_{2,nk} \\ \vdots & \vdots & \ddots & \vdots \\ R_{l,1} & R_{l,2} & \dots & R_{l,nk} \end{pmatrix} \tag{2}$$

The rows represent there are l physical links, and the columns represent $n * k$ paths which are consisted of n VM-VM pairs, each pair has k paths.

We define the following indicative function:

$$R_{i,j} = \begin{cases} 1, & if\ link\, i \in path\, j \\ 0, & if\ link\, i \notin path\, j \end{cases} \tag{3}$$

Application requirements can be denoted by throughput and delay characteristics of the application. The throughput-sensitive parameters are denoted by a vector:

$$\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n)(n \geq 2) \tag{4}$$

and the delay-sensitive parameters are denoted by a vector:

$$\beta = (\beta_1, \beta_2, \dots, \beta_n)(n \geq 2) \tag{5}$$

Based on the requirements, we can define utility functions to reflect the performance of different types of applications:

$$U_i(X_i, X_{-i}) = \sum_{\substack{j=1, \\ m: m \in L(ij)}}^{k} (\alpha_i x_{ij} - \frac{\beta_i x_{ij}}{\gamma_m}), \quad for\ all\ i \in I \tag{6}$$

X_{-i} represents bandwidth allocated to other pairs except for i, $L(ij)$ represents the set of links which are used by VM-VM pair i. in expression (6), the deterministic term $1/\gamma_m$ represents the expected congestion delay on link m from an $M/M/1$ delay function [14]. α_i and β_i reflect the throughput and delay sensitive characteristics of different types of applications respectively. For example, to a throughput-sensitive application (e.g. Map Reduce, video), the parameter α_i is always set larger to reflect that bandwidth affects its utility obviously; to a delay-sensitive application (e.g. electronic transaction), the parameter β_i is usually set larger to reflect delay affect its utility significantly. By choosing suitable α_i and β_i, we can give different types of applications fine-grained service differentiation .

In Table I, the key notations used through the paper are summarized.

TABLE I
KEY NOTATIONS IN THE SYSTEM MODEL

Symbol	Description
l	Number of physical links
n	Number of server -server pairs
c_i	Bandwidth capacity of physical link i
γ_i	Residual bandwidth of physical link i
X_i	Bandwidth allocated to pair i
low_i	Lower bound of bandwidth requirement of pair i
α_i	Throughput-sensitive parameter of pair i
β_i	Delay-sensitive parameter of pair i
U_i	Utility function of pair i

A. The bandwidth allocation scheme

Based on our system model, we can formulate our bandwidth allocation scheme as a multi-objective optimization problem. We present the mathematical model as follows:

$$maximize \quad U_i(X_i, X_{-i}) \tag{7}$$

$$s.t. \quad \gamma_i \geq 0 \tag{8}$$

$$\textstyle\sum_{j=1}^{k} x_{ij} \geq low_i \tag{9}$$

$$0 \leq x_{ij} \leq c_l \tag{10}$$

In this formulation, the objective is to find a optimal bandwidth allocation which can maximize the utility of all applications.

As we have defined routing matrix $R_{l,nk}$ and bandwidth allocation vector $X_{1,nk}$, we can rewrite the expression (8) to (11):

$$R_{l,nk} \times X_{1,nk}^T \leq (c_1, ..., c_l) \tag{11}$$

B. Design discussion

Centralized Management Unit: the centralized bandwidth allocation problem is always subject to NP-HARD, it is hard to find the solution of multi-objective optimization even by using genetic algorithms such as NSGA-II. In order to solve the problem more practically, we use a linearity weighted aggregation method to normalize the multi-objective utility optimization to a general utility optimization.

It means optimize the utility as following:

$$\max f(x) = \sum_{i=1}^{n} \lambda_i U_i(X_1, \ldots, X_n) \tag{12}$$

λ_i can be obtained by computing equation(13):

$$\lambda_i = \sqrt{\alpha_i^2 + \beta_i^2} \quad for\ all\ i \in I \tag{13}$$

We suppose that this method has a great significance in practical, because it is beneficial to allocate the bandwidth to the application which is more helpful to maximize the total utility. And our scheme allocates the bandwidth based on meeting the lower bound requirements of applications first, so do not worry the availability of applications whose λ_i is small.

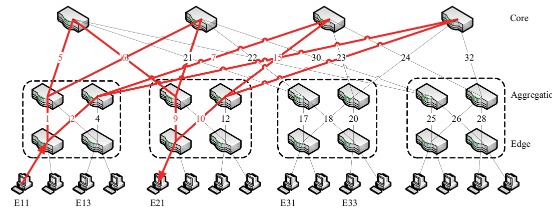

Fig. 2. An example of bandwidth allocation scheme

Hypervisor: according to the allocation result informed by CMU, hypervisors should set an upper limit to each VM-VM pair:

$$up_i = \sum_{j=1}^{j=k} x_{ij} \quad for\ all\ i \in I \tag{14}$$

Since current proposed DCN topologies, including fat-tree, often has the nature of symmetry, in these topologies, we let the bandwidth allocated to each VM-VM pair i on path j are equal:

$$x_{ij} = x_{ij'}(j \neq j') \quad for\ all\ i \in I \tag{15}$$

We can use Equal Cost Multiple Path (ECMP) protocol to allocate bandwidth on each path. In our scheme, the bandwidth capacity is greater than the total used bandwidth on that link, so congestion can be relieved and the negative effects of packets miss-order can be neglected.

Besides, our scheme is suited to arbitrary DCN topologies. We give the corresponding solutions for topologies which are not symmetry. Alike to [5], we use port-switching based source routing as our routing mechanism, hypervisor forwards packets depend on the route decided by CMU in advance. Different from [5], in our scheme, packets should be forwarded to multi-path. It can be implemented by doing a simple modification on hypervisors. According to $X_i = (x_{i1}, x_{i2}, ..., x_{ik})$, the packets can be distributed to path $1 - k$ with relevant weight respectively. Each weight can be calculated by:

$$w_{ij} = \frac{x_{ij}}{\sum_{j=1}^{j=k} x_{ij}} \quad for\ all\ i \in I \tag{16}$$

Hypervisor adds source routing head to each packet with a certain probability decided by weight.

V. NUMERICAL SIMULATION

Based on the model mentioned above, we present a simulation of our DCN bandwidth allocation scheme. In our simulation, we use the topology in Fig.2, in which there are three kinds of applications. In order to simplify the simulation, we assume each application only use one VM-VM pair.

In this topology, there're 32 physical links, the bandwidth capacity of each link is set by 10 Gbps. According to the algorithm mentioned in section III.B, we number each link as shown in Fig.2. There're 3 VM-VM pairs: $E_{11} - E_{21}, E_{21} - E_{31}, E_{13} - E_{34}$, belong to 3 types of applications respectively. To show intuitively, in Fig. 2, we only mark 4 paths used by pair $E_{11} - E_{21}$. The bandwidth allocation vectors can be denoted by:

$$X_i = (x_{i1}, x_{i2}, x_{i3}, x_{i4}) \quad i = 1, 2, 3 \tag{17}$$

The routing matrix can be denoted by:

$$R_{32,12} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{18}$$

In this simulation, we assume the application 1 is a delay-sensitive application, application 3 is a throughput-sensitive application, and application 2 is in between. We manually set application requirements for each pair, the throughput-sensitive parameters are set to:

$$\alpha = (1, 2, 3) \tag{19}$$

The delay-sensitive parameters are set to:

$$\beta = (3, 2, 1) \tag{20}$$

And the lower bound of bandwidth requirements is set to:

$$low = (1, 2, 3) \tag{21}$$

Based on these parameters, we can obtain the utility function for each pair:

$$U_1(X_1, X_2, X_3) = \sum_{\substack{j=1 \\ m \in L(1j)}}^{j=4} (x_{1j} - \frac{3x_{1j}}{\gamma_m}) \tag{22}$$

$$U_2(X_1, X_2, X_3) = \sum_{\substack{j=1 \\ m \in L(2j)}}^{j=4} (2x_{2j} - \frac{2x_{2j}}{\gamma_m}) \tag{23}$$

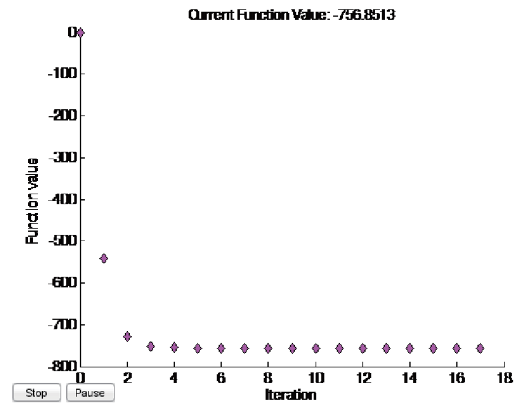

Fig. 3. Iteration convergence of our algorithm

TABLE II
BANDWIDTH ALLOCATION RESULT

$Pairs$	$Path_1$	$Path_2$	$Path_3$	$Path_4$
1	0.2500	0.2500	0.2500	0.2500
2	3.3688	3.3688	3.3688	3.3688
3	4.2699	4.2699	4.2699	4.2699

$$U_3(X_1, X_2, X_3) = \sum_{\substack{j=1 \\ m \in L(3j)}}^{j=4} (3x_{3j} - \frac{x_{3j}}{\gamma_m}) \tag{24}$$

We use Matlab to find the solution of the following mathematical model:

$$maximize \qquad U_i(X_1, X_2, X_3) \tag{25}$$
$$s.t. \quad R_{32,12} \times X_{1,12}^T \leq (10, ..., 10) \tag{26}$$
$$\textstyle\sum_{j=1}^{4} x_{ij} \geq low_i \tag{27}$$
$$0 \leq x_{ij} \leq 10 \tag{28}$$

According to equation (13) and the throughput-sensitive and delay-sensitive parameters of 3 types of applications, we can rewrite (25) to the following single object optimization problem:

$$\max f(x) = \sqrt{10}U_1 + 2\sqrt{2}U_2 + \sqrt{10}U_3 \tag{29}$$

As shown in Fig.3, after 14 times' iterative calculation, we get a good convergence solution. This is the optimal solution in the sense of linearity weighted for multi-objective optimization problem. The final bandwidth allocation result is shown in Table II.

The result shows that our allocation scheme can provide bandwidth guarantee, all allocated bandwidth meet the minimum requirements of all applications (pairs). Meanwhile, the scheme achieves high network utilization obviously, the bandwidth allocated to pair 2 and pair 3 (13.4752 Gbps and 17.0796 Gbps) substantially greater than their minimum requirements (2 Gbps and 3 Gbps), so the network can be effectively utilized. The result also indicates that the extra bandwidth, which exceeds the minimum requirement, is allocated differentially from each other. So services can be fine-grained differentiation by our scheme.

VI. Conclusion

In this paper, we studied the bandwidth allocation problem for data center networks. We propose a bandwidth allocation scheme based on an application utility-based model which is suitable for DCNs. The numerical simulation gives a simple example of our scheme and the result shows that the scheme can provide bandwidth guarantee and fine-grained service differentiation. It also achieves high network utilization by exploiting multi-path feature in DCNs. More importantly, our scheme is scalable and deployable by using centralized management unit combining with edge hypervisor. In the future, we'll fulfill a prototype of our scheme, and apply it in a testbed data center to verify its performance.

Acknowledgment

This work is supported by the National Natural Science Foundation of China(No. 61002011); the Open Fund of the State Key Laboratory of Software Development Environment(No. SKLSDE-2009KF-2-08), BeiHang University; the 973 Program of China(No. 2009CB320505); the 863 Program of China(No.s 2011AA01A102)

References

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, and M. Zaharia, "Above the clouds: A berkeley view of cloud computing," EECS Department, University of California, Berkeley, Tech. Rep., 2009.

[2] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the art of virtualization," *SIGOPS Oper. Syst. Rev.*, vol. 37, no. 5, pp. 164–177, 2003.

[3] S. Kandula, S. Sengupta, A. Greenberg, P. Patel, and R. Chaiken, "The nature of data center traffic: measurements and analysis," ser. Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference. Chicago, Illinois, USA: ACM, 2009, pp. 202–208.

[4] T. Benson, A. Anand, A. Akella, and M. Zhang, "Understanding data center traffic characteristics," *SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 1, pp. 92–99, 2010.

[5] C. Guo, G. Lu, H. J. Wang, S. Yang, C. Kong, P. Sun, W. Wu, and Y. Zhang, "Secondnet: a data center network virtualization architecture with bandwidth guarantees," in *Proceedings of the 6th International COnference*. Philadelphia, Pennsylvania: ACM, 2010, pp. 1–12.

[6] H. Ballani, P. Costa, T. Karagiannis, and A. Rowstron, "Towards predictable datacenter networks," in *Proceedings of the ACM SIGCOMM 2011 conference on SIGCOMM*. Toronto, Ontario, Canada: ACM, 2011, pp. 242–253.

[7] T. Lam, S. Radhakrishnan, A. Vahdat, and G. Varghese, "Netshare: Virtualizing data center networks across services," *University of California, San Deigo, Tech. Rep. CS2010-0957*, 2010.

[8] H. Rodrigues, J. R. Santos, Y. Turner, P. Soares, and D. Guedes, "Gatekeeper: supporting bandwidth guarantees for multi-tenant datacenter networks," in *Proceedings of the 3rd conference on I/O virtualization*. Portland, OR: USENIX Association, 2011, pp. 6–6.

[9] A. Shieh, S. Kandula, A. Greenberg, C. Kim, and B. Saha, "Sharing the data center network," in *Proceedings of the 8th USENIX conference on Networked systems design and implementation*. Boston, MA: USENIX Association, 2011, pp. 23–23.

[10] C. Guo, G. Lu, D. Li, H. Wu, X. Zhang, Y. Shi, C. Tian, Y. Zhang, and S. Lu, "Bcube: a high performance, server-centric network architecture for modular data centers," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 4, pp. 63–74, 2009.

[11] R. Niranjan Mysore, A. Pamboris, N. Farrington, N. Huang, P. Miri, S. Radhakrishnan, V. Subramanya, and A. Vahdat, "Portland: a scalable fault-tolerant layer 2 data center network fabric," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 4, pp. 39–50, 2009.

[12] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta, "Vl2: a scalable and flexible data center network," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 4, pp. 51–62, 2009.

[13] R. Braden, D. Clark, and S. Shenker, "Rfc 1633: Integrated services in the internet architecture: an overview."

[14] U. Javed, M. Suchara, J. He, and J. Rexford, "Multipath protocol for delay-sensitive traffic," in *Communication Systems and Networks and Workshops, 2009. COMSNETS 2009. First International*. IEEE, 2009, pp. 1–8.