# Network Performance-Aware Virtual Machine Migration in Data Centers

Jun Chen, Weidong Liu, Jiaxing Song
Department of Computer Science and Technology
Tsinghua University
Beijing, China
chenjun09@mails.tsinghua.edu.cn
{liuwd, jxsong}@tsinghua.edu.cn

***Abstract*—Virtual machine (VM) consolidation and migration technology in data centers greatly improve the utilization of the server resource. While the previous work focuses on how to use VM migration to balance physical host utilization or optimize energy consumption, little attention has been given to network performance factors, such as link traffic load and inter-traffic between VMs in data centers. In this paper, we present MWLAN (Migration With Link And Node load consideration), a novel automatic data center VM migration system that can detect hotspots (e.g., network congestion and physical host over-loaded) and dynamically remap VMs to improve the network performance. The VM migration approach proposed in MWLAN can efficiently balance the network link load and relieve the local data center network congestion as well as considering physical host constraints. Moreover, experimental evaluations indicate that the proposed approach reduces the packet loss by up to 50% and improves the average application TCP transfer rate by up to 24% compared to the other approaches when the data center network overloaded.**

***Keywords-virtualization; virtual machine migration; data center; load balancing***

## I. INTRODUCTION

With the development of technology, virtualization has been widely used in data centers. It allows a single physical host to run multiple isolated virtual machines. When a physical host is overloaded, virtual machine migration can dynamically remap virtual machines onto physical hosts in data centers, which greatly improves physical host resource utilization. At the same time, network scalability is becoming more and more crucial in data center network system. Many new network architectures [2][3] have been proposed for data centers to solve the network problems. As the server virtualization on data centers, the VMs placed in data center physical hosts are applications or application components (multi- tier applications). There are usually high traffic rate and increasing trend towards more communication due to the inherent coupling among VMs (e.g., scientific computing, web search, MapReduce). The VMs arrive/depart dynamically and their location is not fixed. In such environments, VMs with large communication or belonging to the same application tier are very likely to be scattered into different network segments. We call it service fragmentation, which consumes large inter-node bandwidth. The research [15] shows that service fragmentation can heavily affect the data center network performance. Thus, how to schedule and place the VM to improve the data center network performance is a meaningful topic.

However, in recent years, many work focus on using virtual machines (VMs) consolidation and migration to improve the efficiency of physical host or power management in data centers. Little attention has been given to the network performance influence of VM migration in data centers.

In this paper, we present a novel migration system, MWLAN (Migration With Link And Node load consideration), a dynamic migration scheduling system in data centers. MWLAN collects the load information on physical host and switch links, detects and finds hotspots. After that it chooses a VM candidate and a physical host candidate for VM migration, taking the underlying data center network performance factors into count, as well as the physical host constraints. However, the VMs migration problem with resource constraints on physical node and link can be reduced to virtual network embedding/mapping (*VNE*) problem which is proven to be NP-complete [4]. In this paper, a heuristic algorithm is proposed to solve the migration problem efficiently. The ultimate goal is to balance the network traffic load and improve the network resource utilization while satisfying VMs and physical host resource constraints in data centers. Furthermore, the experiment results demonstrate that the proposed approach reduces the packet loss by up to 50% and improves the average application transfer rate up to 24% compared to the other approach when the data center network overloaded according to scheduling 10% VMs.

**Our contributions can be summarized as follows:**

- We address the problem of network link load dynamic adaption and formulate the cost of network link load in data centers in order to avoid network congestion or overload.
- We propose a novel VM dynamic migration idea by efficiently utilizing network resources as well as considering physical host constraints.
- We evaluate the proposed algorithms by simulators and the results prove that they can significantly

relieve network congestion and improve the traffic rate when network overloaded.

The rest of this paper is organized as follows. Section II provides some background and gives an overview of the migration system MWLAN. Section III presents our core system architecture of MWLAN. In Section IV, we evaluate the proposed methods using simulations. Then Section V discusses the related work. Finally, Section VI presents our conclusion and future work.

## II. BACKGROUND

The existing data center VM migration approaches are used to eliminate the overloaded physical host, which move a virtual machine from the overloaded physical host to another underloaded one. This migration policy can balance the utilization of physical host resource. But no one considers using VM migration to balance the data center network link traffic load and prevent network performance degradation. This paper designs a data center virtual machine migration management system MWLAN. MWLAN is used in virtualized data center. Generally, a virtualized data center is composed by network and physical hosts (or server). The interconnected switches formulate the data center network [2][3], while the physical hosts are connected by data center network. One physical host can hold one or more VMs which are allocated some parts of physical host resource, such as CPU, memory. Each VM runs an application or an application component (multi-tier application). All storage is thought to be on a network file system (NFS) or a storage area network (SAN), thus, MWLAN can avoid storage migration.

More specifically, MWLAN has full knowledge of the network topology, network configuration (routing info), the switch link bandwidth, the physical host capacity and the mapping of applications to physical host. By taking a global view of routing and VM traffic demands, MWLAN can identify the load of physical host and the switch link in data centers. If a hotspot occurs (e.g., network congestion and physical host overloaded), MWLAN can use VM migration to balance the overloaded resources (e.g., physical host or links). Figure 1 shows the virtualized data center and MWLAN.

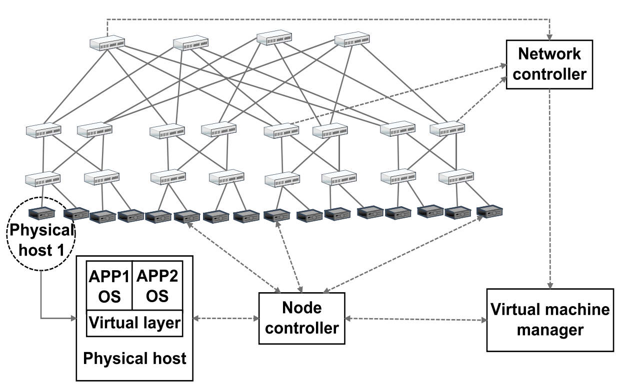


Figure 1. The virtualized data center and MWLAN architecture.

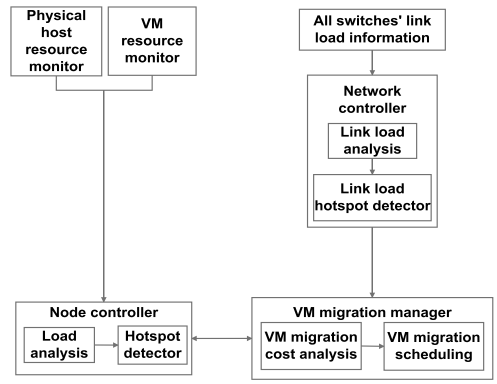


Figure 2. The MWLAN architecture.

MWLAN is consisting of three components: Node Controller, Network Controller and VM Migration Manager. Node Controller is responsible for gathering VM resource usage statistics on each physical host and VMs, doing demand estimation (physical host resource demand and bandwidth demand) and detecting physical host hotspot. Network Controller periodically gathers link bandwidth usage statistics of data center network and the routing info, and then does the link load hotspots detecting process. VM Migration Manager is responsible for choosing the migration VM candidate and the destination physical host candidate. Therefore we propose the MWLAN architecture depicted in Figure 2, the principal components and their interplay are described in more detail in the following architecture section.

## III. MWLAN SYSTEM ARCHITECTURE

The below section will discuss the detail function of MWLAN's components which can be divided into four steps. First, host and network resource usage monitoring in date centers, such as physical host usage, VM resource usage and the traffic load at the switches. Next, it describes the hotspot detection. And then, demand estimation and VM migration cost analysis. Finally, VM migration schedule.

### A. *Monitoring*

Monitoring is not only responsible for tracking the resource usage of physical host and VMs, but also gathering the link bandwidth consumed information of switches in data centers. Thus, monitoring is composed by two parts: host monitoring and network monitoring.

Host monitor runs on each physical host and VM. It gathers the host resource usage, such as the CPU usage, the transfer data rate of VMs. As shown in Figure 2, the node controller gathers all hosts' resource usage information from host monitor.

Network monitoring is running on each switch in data centers. It periodically measures the link load of the switch (such as switch logs) and sends the link load information to the network controller.

 

### B. Hotspot Detection

Hotspot detection is used to find out the hotspot on physical host and switch link. As shown on Figure 2, the host controller has a hotspot detector which is responsible for detecting hotspot on physical host. The network controller has a hotspot detector which is responsible for detecting hotspot on the switch link in data centers.

#### 1) Host Hotspot Detection

The physical host load metric contains CPU, memory, network facts. A physical host may be overloaded on one or more facts. So, we use $vol_{node}$ [5] as the quantification of the physical host load. If the physical resources are more overloaded, the $vol_{node}$ will be higher.

$$Vol_{node} = \frac{\omega_1}{1-cpu} * \frac{\omega_2}{1-memory} * \frac{\omega_3}{1-net_{node}} \quad (1)$$

$\omega_i$: the weight of CPU, memory and network load.
*cpu*: the physical host CPU utilization.
*memory*: the physical host memory utilization.
$net_{node}$: the physical host network port utilization.
*n*: the continues observation times.
*k*: the threshold of overload times.
$\lambda_{node}$: the threshold of $vol_{node}$.

If there are more than k times $vol_{node}>\lambda_{node}$ in the last n detections, the physical host may be thought to be overloaded [5], a hotspot is detected. Then, it schedules the VM migration manager to do a VM migration to eliminate the hotspot.

#### 2) Network Hotspot Detection

The network resource of a data center is the switches' link bandwidth. Thus, the utilization of the link bandwidth is the load of each switch link traffic load. $vol_{net}$ is used to be the quantification of switch link load. If the utilization of the link is high, the $vol_{net}$ will be high.

$$Vol_{net} = \frac{\alpha_{link}}{1-net_{link}} \quad (2)$$

$\alpha_{link}$ : The weight of switch link, if some of the switch link is much more valuable (such as the bottleneck link of the data center network), the weight $net_{link}$ of this link will be bigger.

$net_{link}$: The link bandwidth utilization.

Similar to host hotspot detection, a network hotspot is flagged only if $vol_{net}$ exceeds a threshold $\lambda_{net}$ for a sustained time *k* in the recent *n* time observations.

### C. Demand Estimation and VM Migration Cost Analysis

As the VM's current used resource may not reflect the actual demand, MWLAN must estimate the VM's actual resource demand before migration. There are many multi-tier applications models to estimate the multi-tier application resource demand. The queuing model [10] is used as the basic of the VM demand estimation. By using the monitored information of application VMs (Gray-box monitoring [5]) and the model for multi-tier applications, MWLAN can estimate the VM physical resource demand (e.g., CPU demand) and VM's actual bandwidth demand.

VM migration scheduling is responsible for choosing which VM to migrate and which physical host to hold the migration VM. And our ultimate goal is to balance the network traffic load and improve the network resource utilization.

If a VM is moved from one physical host to another host, the flows which related to the migration VM will switch too. So when we schedule VMs, a key challenge is to estimate the migration impact to the traffic loads on links. To solve this problem, the system quantifies the impact of the virtual machine migration on network. It considers the bandwidth consumed and the link load by the flows related to the VM before and after the migration.

The $VM_w$ consumed network resource in data centers can be defined as:

$$Cost = \sum_{\forall VM_p \in \{\forall VM_i | C_E(VM_w, VM_i) != 0\}} \sum_{\forall e \in E(VM_w, VM_p)} C_E(VM_w, VM_p) \quad (3)$$

The variables in the Equation are defined as Table I shown.

TABLE I. THE DEFINITION OF VARIABLES IN THE EQUATION

| Variable | Description |
|---|---|
| $C_E(VM_w,VM_P)$ | The transfer data rate between $VM_w$ and $VM_p$. |
| $C_N(VM_w)$ | The amount of physical resource which is allocated to $VM_w$ on physical host. |
| $E(VM_w,VM_P)$ | The switch link set of transfer data path between $VM_w$ and $VM_p$. |
| $E'(VM_w,VM_p)$ | The switch link set of transfer data path between $VM_w$ and $VM_p$ if $VM_w$ is migrated to physical host $PM_k$. |
| $R_E(e)$ | The remaining available bandwidth on link *e*. |
| $R_N(PM_w)$ | Physical host $PM_w$ remaining available resource. |
| $\alpha_e$ | The weight of switch link *e*. |
| $\beta$ | The weight of physical host. |
| $\delta$ | Constant, to make ensure the denominator is bigger than zero. |
| $\{\forall VM_i \mid C_E(VM_w,VM_i)!=0\}$ | The VM set which has network traffic with $VM_w$. |

Since our objective is to balance the link traffic load, the utilization of links should also be taken into account. So if the $VM_w$ is migrated from physical host $PM_w$, the effect to the network traffic load can be defined as:

$$Revenue(VM_w) = \sum_{\forall VM_p \in \{\forall VM_i | C_E(VM_w, VM_i) != 0\}} \sum_{\forall e \in E(VM_w, VM_p)} \frac{\alpha_e}{R_E(e)+\delta} C_E(VM_w, VM_p) \quad (4)$$

 ISBN: 978-1-61208-216-5 

*Revenue* $(VM_w)$ considers both the consumed network resource of $VM_w$ and the utilization of related switch links. If moving $VM_w$ away from physical host $PM_w$, the traffic load of the switch links used by $VM_w$ will be relieved. So (4) denotes the positive effect to the data center network by moving $VM_w$ away from original physical host.

Similarly, the network cost of placing a VM $VM_w$ on physical host $PM_k$ can be defined as:

$$Cost(VM_w) = \sum_{\forall VM_p \in \{\forall VM_i | C_E(VM_w, VM_i) != 0\}} \; \sum_{\forall e \in E^{'}(VM_w, VM_p)} \frac{\alpha_e}{R_E(e)+\delta} C^{'}_E(VM_w, VM_p) \quad (5)$$

For each $VM_w$, if it is moved from the original physical host to a candidate host, we denote the benefit of this schedule by *Benefit*$(VM_w)$.

$$Benefit(VM_w) = Revenue(VM_w) - Cost(VM_w) \quad (6)$$

And if taking physical host load into consideration, we define the benefit of a VM $VM_w$ migration from physical host $PM_w$ as:

$$Revenue^{'} = \sum_{\forall VM_p \in \{\forall VM_i | C_E(VM_w, VM_i) != 0\}} \; \sum_{\forall e \in E(VM_w, VM_p)} \frac{\alpha_e}{R_E(e)+\delta} C_E(VM_w, VM_p) + \frac{\beta}{R_N(PM_w)+\delta} C_N(VM_w) \quad (7)$$

Similarly, the cost of a VM $VM_w$ is placed on physical host $PM_k$ can be defined as:

$$Cost^{'} = \sum_{\forall VM_p \in \{\forall VM_i | C_E(VM_w, VM_i) != 0\}} \; \sum_{\forall e \in E^{'}(VM_w, VM_p)} \frac{\alpha_e}{R_E(e)+\delta} C^{'}_E(VM_w, VM_p) + \frac{\beta}{R_N(PM_k)+\delta} C^{'}_N(VM_w) \quad (8)$$

### D. VM Migration Schedule

According to the above migration *cost* and *revenue* equations, the intuitive migration manager policy proceeds as follows: At first, compute the migration revenue of each candidate VM which is located on the overloaded physical host or which traffic flows are forwarded by the overloaded link. After that, referring to the above migration *revenue* (4), we sort the VM migration *revenue* in decreasing order. The policy chooses the candidate VM of the maximum *revenue* as the one to migrate. By considering VMs in *revenue* order, the algorithm attempts to migrate the VM which has the biggest potential to relieve the link load and the bandwidth cost. And then, according to the above migration *cost* (5), the migration manager first computers the candidate VM migration *cost* on each underloaded physical host. And again we sort the VM migration *cost* of the each physical host in increasing order. The policy chooses the minimize *cost* physical host as the destination physical host for the candidate VM, which also aims to minimize the network cost. The main steps of this strategy are listed in Algorithm 1. The complexity of the Algorithm 1 is *O (max (m,n))*, where m denotes the candidate VM number, and n denotes the number of physical host which can hold the migration VM.

**Algorithm 1** virtual machine migration (**MWLAN1**)
**Require**: the overload physical machine(PM) $PM_w$

```
For each VM_i ∈ PM_w
    R(VM_i) = Revenue(VM_i,PM_w)
end for
//Note: Revenue computed by (4)
sort VM_i ∈PM_w in decreasing order Revenue (VM_i))
for each VM_i ∈PM_w in decreasing order Revenue (VM_i))
    VM_migration= VM_i, PM_dest = NULL
    Min_cost = inf
    for each PM_j in a data center
        if (!check_pm_constrain (PM_j, VM_migration))
            continue // pm can't hold the vm
        end if
        //Note: Cost computed by (5)
        cost(PM_j)= Cost (VM_migration, PM_j)
        if (cost(PM_j) <Min_cos)
            PM_dest = PM_j
            Min_cost = cost(PM_j)
        end if
    end for
    if(PM_dest == NULL)
        continue
    else
        break
    end if
end for
if(PM_dest == NULL)
    no physical machine can hold a migration VM
    return
else
    return { VM_migration , PM_dest }
end if
```

While the Algorithm 1 takes into account both migration *revenue* and *cost*, it can't make sure that the migration gets to maximum benefit which is defined on (6).

The Algorithm 2 merges the process of VM candidate and destination physical host choosing. As the total migration should both consider the *revenue* and *cost*, the Algorithm 2 chooses the migration VM and destination physical host which maximizes *benefit* (6) among all candidate VMs on overload PM and all candidate physical host in data centers. The complexity of algorithm 2 is *O (mn)*, where m denotes the candidate VM number, and n denotes the number of physical host which can hold the migration VM.

**Algorithm 2** virtual machine migration (**MWLAN2**)
**Require**: the overload physical machine(PM) $PM_w$

```
Max_benefit =0 ,Total_benefit = 0
Current_benefit = 0, Min_cost = inf
Current_pm = NULL,VM_migration= NULL
PM_dest = NULL
for each VM_i ∈PM_w
        //Note: Revenue computed by (4)
        R(VM_i) = Revenue (VM_i,PM_w)
        Current_pm = NULL
        for each PM_j in a data center
                If(!check_pm_constrain(PM_j,VM_i))
                        continue  // pm can't hold the vm
                end if
                //Note: Cost computed by (5)
                Current_cost = Cost (VM_i , PM_j)
                If(Current_cost < Min_cost)
                        Min_cost = Current_cost
                        Current_pm = PM_j
                end if
        end for
        if(Current_pm = =NULL)
                continue
        else if( (R(VM_i)- Min_cost) > Max_benefit )
                VM_migration= VM_i
                PM_dest = PM_j
        end if
end for
if(PM_dest == NULL)
        no physical machine can hold a migration VM
        return;
else
        return { VM_migration , PM_dest }
end if
```

If we also consider the physical host load balancing as well as link load balancing, we can use (7)(8) as the VM migration revenue and cost to replace (4)(5).

## IV. EVALUATION

This section describes the evaluation of MWLAN and other migration schemes on simulated data center. The goal of these tests is to compare data center link load on different migration schemes and analyze the impact of different migration schemes on application's TCP transfer rate. The simulated data center is implemented by using ns-3 simulator. Ns-3 [1][11] is a discrete-event network simulator and used in lots of research work [12][13]. What's more, ns-3 is free software, licensed under the GNU GPLv2 license.

### A. Evaluation Setup

We use ns-3 to generate a three-layer tree structure of the data center network. Each leaf node is a physical host. Each non-leaf node is a 10-port switch which is connected with sub-node. This data center network has 1 0-level switch which link bandwidth is 5MB/S, 10 1-level switches which link bandwidth is 1MB/S, 100 physical hosts, so the 0-level switch will be the bottleneck of data center network. In order to compare the efficiency of migration schemes on different data centers' link load, we increase the number of VMs placed in the data center from 0 to 360. All the VMs are 2 tier multi-tier application components. Each VM only transfers data with the other VM which belongs to the same multi-tier application. The default transfer protocol is TCP. The detail simulation parameters are noted in Table II.

TABLE II. PARAMETER FOR SIMULATIONS

| Variable | Distribution | Mean | Var |
|---|---|---|---|
| Capacity (PM) | Normal | 1.8 | 0.1 |
| Demand (VM) | Normal | 0.2 | 0.1,0.2 |
| Rate (VM) | Normal | 0.2,0.4 | 0.1 |
| Arrival of VMs | Poisson | 20(s) | 20(s) |
| The initial placement of arrived VMs | Random,<br>Same switch,<br>Different switch | | |
| Num of VMs | 0-360 | | |
| VM migration schedule interval | 200(s) | | |
| Data center network topology | Tree | | |

Because the efficiency of migration schemes may vary with different traffic patterns caused by the initial placement of VMs before migration, we run the compared test on three different VM initial placement patterns. In the first pattern, the initial placement of arrived VM is random (Random Pattern). And the VMs which have traffic are placed in the same 1-level switch in the second pattern (Same Pattern). And in the last pattern, the VMs which have traffic are placed in different 1-level switches (Different Pattern).

The benchmark tests are running as follows: we assume the VM requests arrive in a Poisson process with an average rate of 1 VMs per 20 seconds units. Each VM sends data to another VM using TCP protocol. VM migration occurs periodically every 200 seconds. This configuration can make sure the percentage of the migration VM is about 10%. Considering the VM migration cost, 10% is an appropriate migration proportion. The experiment lasts until the number of VMs larger than 360 in the data center. We implement MWLAN 1 and MWLAN 2 which are presented in Section III. The test compares MWLAN 1 and MWLAN 2 with previous migration scheme Sandpiper [5] which moves the VM from the most overloaded physical host to the least overloaded physical host. All VM migration schemes make sure total load of VMs on a physical host that doesn't larger than its capacity. In this paper, the experiments employ several network performance metrics: the average TCP transfer rate and the total link packet loss in the data center.

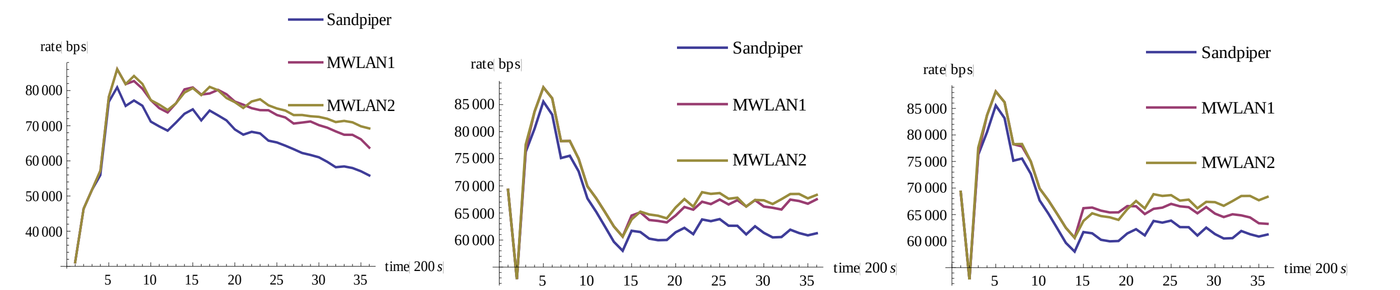


Figure 3(a) Random Pattern Figure 3(b) Same Pattern Figure 3(c) Different Pattern

Figure 3. Average VM transfer rate in three initial VM placement patterns

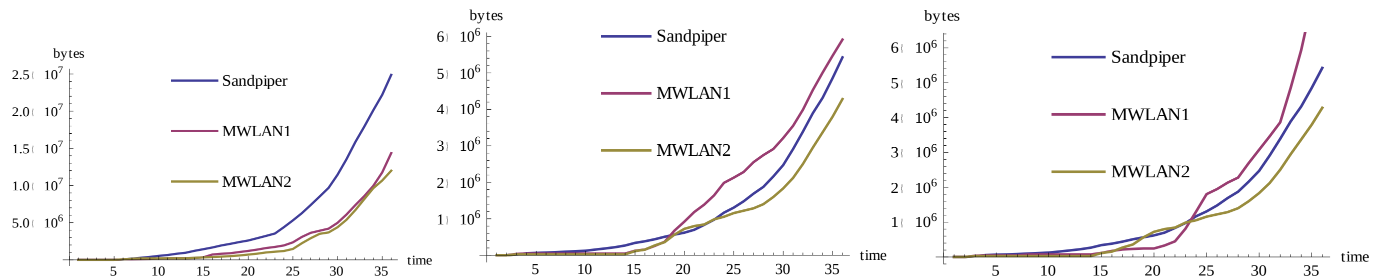


Figure 4(a) Random Pattern Figure 4(b) Same Pattern Figure 4(c) Different Pattern

Figure 4. Total packet loss in three initial VM placement patterns

### *B. Evaluation Results and Analysis*

Figure 3 shows the application average TCP transfer rate of Sandpiper and MWLAN as time changes on different VM placement patterns. The result indicates that the application performance of MWLAN2 is better than the other scheme as the VM load increasing. The average improvement of application rate is up to 24% compared to Sandpiper. As shown in Figure 3, the average TCP rate is nearly the same in the beginning. And as the VM load increases, the TCP average rate differs to each other for three VM migration approaches. The traffic rate declines more obviously when using Sandpiper compared to our approaches. The reason is that there is no network congestion when the traffic load is not heavy in data cent network. So the VMs can achieve the demand TCP rate. But as the VM load is increasing, the link traffic load is becoming heavier. When network congestion occurs, the TCP rate decreases, as what we see in Figure 4. And MWLAN1 and MWLAN2 consider the link load cost. So they will move the traffic flows from the loaded links to the underloaded links by using VM migration or move the VMs with heavy traffic near to each other for saving link bandwidth cost. Thus MWLAN1 and MWLAN2 not only eliminate the local traffic congestion but also improve the utilization of network resources. These two factors make MWLAN 1 and MWLAN 2 have better network improvement compared to Sandpiper. The Figure 3 also indicates that MWLAN2 has better network performance improvement than MWLAN1. The reason is that MWLAN1 consider the migration revenue and cost separately, it can't make sure the VM migration achieves the maximum benefit, while MWLAN2 always chooses the VM and destination physical host which can get the maximum benefit.

The experiments also make a comparison on the link load when using different VM migration schemes in the data center. We use the packet loss amount as a comparison object. The link packet loss amount can reflect the load of link traffic in the data center. It can be seen from the Figure 4, MWLAN2 outperforms MWLAN1 and Sandpiper, decreasing total link packet loss up to 50% compared to Sandpiper. It reflects that MWLAN2 policy can be more efficient to avoid network congestion in contract to the other policies. Because MWLAN1 can't get maximum migration benefit, it is not as good as MWLAN2. The MWLAN1 only takes load revenue into account when it chooses candidate VM to migrate, the VM which has high migration revenue may also have high migration cost. As a result, MWLAN1 may burden link load and cause network congestion. On the other hand, MWLAN2 always choose the candidate VM and physical host which can get maximum migration benefit. Thus, MWLAN2 can find the best approach to change link load dynamic to avoid and relieve network congestion.

## V. RELATED WORK

As VM migration is transparent to the application [14][16][17], virtual machines consolidation and migrations based on data centers have attracted significant attention in recent years [5][6], many works focus on improving the efficiency of physical host or power management in data centers.

The work in [6] employs dynamic VM consolidation to reduce the number of working physical host in data centers. Wood et al. implement a system that automates the task of monitoring and detecting hotspots, eliminating physical host hotspots by using VM migration [5]. However, this proposed migration algorithm only considers physical host and virtual machine node-resource load (such as CPU, memory)), which ignores the impact of inter-communication between virtual machines and the data center network factors (link bandwidths, the distance between physical machine). Verma et al. [7] discuss the issue between the physical resource utilization and the data center power consumption. It analyzes the application workload and makes consolidation for power saving. Again, these above approaches do not take the effects on underlying network traffic and link load into account when doing VM consolidation and migration in data centers.

Recent proposals [8][9] for VM placement and migration consider network traffic among virtual machines, But they only consider the total transfer data between virtual machine and the distance between physical machines when doing migration. This network factor is too coarse-grained to effectively use the data center network resources, while our migration system considers not only the traffic among VMs but also link traffic load of data centers.

In contrast to our work, none of the approaches mentioned above addresses the problem of network link load dynamic adaption in order to avoid network congestion or overload.

## VI. CONCLUSION AND FUTURE WORK

Previous VM consolidation and migration strategy mainly focus on the physical host resource utilization or physical host load balancing, but ignore the factors of data center network and the traffic between VMs. As the network performance is becoming more and more important in data centers, how to use VM migration to improve the data center network traffic load is a meaningful research topic. This paper proposes a novel migration strategy MWLAN. It quantifies the benefit of VM migration and the cost of VM placement to the network link load in data centers. This migration strategy takes the data center network link load and link bandwidth cost factor into account to solve the migration problem efficiently. What’s more, the experimental results demonstrate that MWLAN has better network performance compared to the other schemes. MWLAN not only reduces data center network congestion but also improves the application transfer data rate. For future work, we look forward to implementing and evaluating our scheme on different kinds of data center network. Moreover, we plan to coordinate the VM placement and VM migration policy for network load balancing in data centers.

## REFERENCES


[1] T. R. Henderson, M. Lacage, and G. F. Riley. Network Simulations with the ns-3 Simulator. Demo paper at ACM SIGCOMM'08, August 2008.

[2] Radhika Niranjan Mysore, Andreas Pamboris, Nathan Farrington, Nelson Huang, Pardis Miri, et al. PortLand: a scalable fault-tolerant layer 2 data center network fabric, Proceedings of the ACM SIGCOMM 2009 conference on Data communication, August 16-21, 2009, Barcelona, Spain

[3] Chuanxiong Guo, Guohan Lu, Dan Li, Haitao Wu, Xuan Zhang, et al. BCube: a high performance, server-centric network architecture for modular data centers, Proceedings of the ACM SIGCOMM 2009 conference on Data communication, August 16-21, 2009, Barcelona, Spain

[4] D.G. Andersen. Theoretical approaches to node assignment. http://www.cs.cmu.edu/~dga/papers/andersen-assign.ps 2002.

[5] T. Wood, P. J. Shenoy, A. Venkataramani, and M. S. Yousif. Black-box and gray-box strategies for virtual machine migration. In Proc. of the 4th Symposium on Networked Systems Design and Implementation (NSDI). USENIX, 2007.

[6] Fabien Hermenier, Xavier Lorca, Jean-Marc Menaud, Gilles Muller, and Julia Lawall. Entropy: a consolidation manager for clusters, Proceedings of the 2009 ACM SIGPLAN/SIGOPS international conference on Virtual execution environments, March 11-13, 2009, Washington, DC, USA.

[7] Verma, P. Ahuja, and A. Neogi. pMapper: Power and migration cost aware application placement in virtualized systems. Technical report, IBM, 2008.

[8] X. Meng, Y. Pappas, and L. Zhang. Improving the scalability of data center networks with traffic-aware virtual machine placement. IEEE INFOCOM, 2010.

[9] V. Shrivastava, P. Zerfos, K. won Lee, H. Jamjoom, Y.-H. Liu, and S. Banerjee. Application-aware virtual machine migration in data centers. In Proc. of IEEE INFOCOM Mini-conference, Apr. 2011.

[10] B. Urgaonkar, G. Pacifici, P. Shenoy, M. Spreitzer, and A. Tantawi. An Analytical Model for Multi-tier Internet Services and Its Applications. In Proc. of the ACM SIGMETRICS, Banff, Canada, June 2005.

[11] http://www.nsnam.org/

[12] Jahanzeb Farooq and Thierry Turletti. An IEEE 802.16 WiMAX module for the NS-3 simulator, Proceedings of the 2nd International Conference on Simulation Tools and Techniques, March 02-06, 2009, Rome, Italy.

[13] Thomas R. Henderson, Sumit Roy, Sally Floyd, and George F. Riley. ns-3 project goals. Proceeding from the 2006 workshop on ns-2: the IP network simulator, October 10-10, 2006.

[14] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield. Live migration of virtual machines. In *Proc. NSDI '05*, May 2005.

[15] Y. Zhang, A. Su and G. Jiang. Evaluating the Impact of Datacenter Network Architectures on Application Performance in Virtualized Environments, Proceedings of 18th IEEE International Workshop on Quality of Service (IWQoS), 2010.

[16] M. Nelson, B. Lim, and G. Hutchins. Fast Transparent Migration for Virtual Machines. In Proc. USENIX 2005.

[17] Sherif Akoush, Ripduman Sohan, Andrew Rice, An-drew W. Moore, and Andy Hopper. Predicting the performance of virtual machine migration. Modeling, Analysis, and Simulation of Computer Systems, International Symposium on, 0:37–46, 2010.

[18] N.M.M.K. Chowdhury, M.R. Rahman, and R. Boutaba. Virtual network embedding with coordinated node and link mapping, IEEE INFOCOM, 2009.