On the Parallel Reconfiguration of Virtual Networks in Hybrid Optical/Electrical Datacenter Networks

Sicheng Zhao†, Xiaqin Pan‡, and Zuqing Zhu†
†School of Information Science and Technology, University of Science and Technology of China, Hefei, China
‡Engineering Technology Center, Southwest University of Science and Technology, Mianyang, China
†Email: {zqzhu}@ieee.org

Abstract—Recently, hybrid optical/electrical datacenter networks (HOE-DCNs) have been considered as a promising DCN architecture, because they merge the merits of electrical packet switching (EPS) and optical circuit switching (OCS). This paper considers the reconfiguration of virtual networks (VNTs) in an HOE-DCN to address the dynamic nature of emerging network services. Specifically, we study the problem that given the original and new virtual network embedding (VNE) schemes of several VNTs, how to schedule parallel virtual machine (VM) migrations in batches to realize the VNT reconfiguration within the shortest time. We design two algorithms to reconfigure the inter-rack network in an HOE-DCN in steps, schedule VMs to migrate accordingly, and allocate bandwidth to the VM migrations. The first algorithm uses the one-shot approach, where all the VM migrations are conducted in parallel within the shortest possible time. We formulate a linear programming (LP) to solve the bandwidth allocations in it exactly. Next, to relieve the bandwidth competition introduced by the one-shot approach, we propose the second algorithm by leveraging the multi-shot approach, i.e., invoking multiple batches of parallel VM migrations such that the reconfiguration time can be further reduced. Extensive simulations verify the effectiveness of our proposals.

Index Terms—Hybrid optical/electrical datacenter networks, Network virtualization, VM migration, Parallel reconfiguration.

I. INTRODUCTION

Over the past decades, due to the raising of 5G, Big Data analytics, and other bandwidth-hungry applications [1], the Internet traffic that is related to datacenters (DCs) has been increasing exponentially [2]. Therefore, DC networks (DCNs) are facing substantial challenges from architecture scalability, energy efficiency, and management agility [3]. This motivates people to not only reconsider the architecture of DCNs, but also design novel network orchestration techniques for DCNs. From the perspective of DCN architecture, one of the most promising improvements is to consider the hybrid optical/electrical DCN (HOE-DCN) [4, 5], which inserts one or more optical switches in the inter-rack network of a DCN and combines optical circuit switching (OCS) together with electrical packet switching (EPS). This is because the advantages of OCS, especially those on bandwidth capacity and energy efficiency [6–11], and can be integrated seamlessly with the benefits of EPS, to make the DCN more scalable, cost-effective, and adaptive for emerging applications with various quality-of-service (QoS) requirements.

On the other hand, network orchestration can leverage new techniques, such as network virtualization [12–14], software-defined networking (SDN) [15–19], and machine learning based artificial intelligence (AI) [20–23], to allocate the IT and bandwidth resources in a DCN in the timely, flexible and application-aware manner. For instance, in [22, 24], we combined SDN, AI and network virtualization to experimentally demonstrate the knowledge-defined network orchestration (KD-NO) based on predictive analytics for HOE-DCNs, which not only realized timely and precise resource allocations to support highly-dynamic DC applications (e.g., Hadoop MapReduce), but also further squeezed the well-known energy-latency tradeoff. Specifically, each DC application was treated as a virtual network (VNT), which consists of virtual machines (VMs) for computing/storage tasks and virtual links (VLs) to enable the data transfers among the VMs, and we designed KD-NO schemes to adjust the virtual network embedding (VNE) scheme of the VNT adaptively to avoid the performance degradations due to resource competition.

Although the proposals regarding KD-NO in HOE-DCNs [22, 24] were promising, the VNT reconfiguration involved in them has to be further studied to ensure the scalability and practicalness. More specifically, two additional problems should be tackled, which are 1) how to calculate the reconfiguration schemes of VNTs based on the network state in an HOE-DCN, and 2) given a VNE reconfiguration scheme, how to design the procedure to accomplish it quickly in the HOE-DCN. We have considered the first problem in [25], while to the best of our knowledge, the second one has not been addressed in the literature yet.

Note that, given the original and new VNE schemes of a VNT, we need to reconfigure the mapping schemes of both the affected VMs and VLs [26, 27]. This means that certain VMs need to be migrated to new servers, while all the affected VLs need to be embedded in the EPS-and-OCS-combined inter-rack network according to the new VNE scheme. It is known that in DCNs, a VM migration usually takes tens of seconds or even minutes [28], which will be much longer than the time used for remapping VLs. Hence, for the second problem about the reconfiguration procedure, we can ignore the latency of VL remappings, but concentrate on the scheduling of VM migrations to minimize the reconfiguration latency.

Although the design of VNT reconfiguration procedure can be reduced to solving the scheduling of related VM migrations, the second problem is still fundamentally different from the conventional scheduling of VM migrations in DCNs [29, 30]. The reasons are two-fold. Firstly, even though we ignore the
we conduct numerical simulations to evaluate our proposals. Ration algorithms are designed in Section III. In Section IV, provides the problem description. The two VNT reconfigu-
latency can be further reduced. Extensive simulations veri fy it exactly to obtain the shortest migration time. The second
shot approach as a linear programming (LP) model, and solve
batches of parallel VM migrations and reconfiguration of
related VLs. We formulate the bandwidth allocation in the on e-
shortest possible time, and then reconfigure the OXC and
parallel VM migration algorithms to reconfigure the inter-rack
VNT reconfiguration within the shortest time. We design two
schedule parallel VM migrations in batches to realize the
original and new VNE schemes of several VNTs, how to
VM migrations even more complex.

In this paper, we investigate the problem that given the
original and new VNE schemes of several VNTs, how to
schedule parallel VM migrations in batches to realize the
VNT reconfiguration within the shortest time. We design two
parallel VM migration algorithms to reconfigure the inter-rack
network in an HOE-DCN. This makes the scheduling of
VM migrations even more complex.

In this paper, we investigate the problem that given the
original and new VNE schemes of several VNTs, how to
schedule parallel VM migrations in batches to realize the
VNT reconfiguration within the shortest time. We design two
parallel VM migration algorithms to reconfigure the inter-rack
network in an HOE-DCN in steps, schedule VMs to migrate
accordingly, and allocate bandwidth to the VM migrations.
The first algorithm is referred to as the one-shot approach,
where we first conduct all the VM migrations in parallel within
the shortest possible time, and then reconfigure the OXC and
related VLs. We formulate the bandwidth allocation in the one-
shot approach as a linear programming (LP) model, and solve
it exactly to obtain the shortest migration time. The second
algorithm (i.e., the multi-shot approach) leverages multiple
batches of parallel VM migrations and reconfiguration of
OXC and related VLs to relieve the bandwidth competition
caused by the one-shot approach, and thus the reconfiguration
latency can be further reduced. Extensive simulations verify
the effectiveness of our proposals.

The rest of the paper is organized as follows. Section II
provides the problem description. The two VNT reconfigu-
ration algorithms are designed in Section III. In Section IV,
we conduct numerical simulations to evaluate our proposals.
Finally, Section V summarizes the paper.

II. PROBLEM DESCRIPTION

A. Network Model

We model the topology of the inter-rack network in an HOE-
DCN as a graph \( G(V_s, E_s) \), where \( V_s \) and \( E_s \) are the sets of
substrate nodes (SNs) and substrate links (SLs) for network
virtualization, respectively. Here, each SN \( v_s \in V_s \) represents
a server rack that consists of a top-of-rack (ToR) switch and
a few servers, and each SL \( e_s \in E_s \) can be either an EPS-
based or an OCS-based network connection. In the HOE-DCN,
each pair of ToR switches are constantly connected through its
EPS-based part (i.e., a hierarchical topology based on Ethernet
switches), while they can also talk through the OCS-based part
if the OXC is properly configured. For instance, the OXC in
Fig. 1 is configured to enable OCS-based connections between
rack pairs 1-3, 2-5, 4-6, and 7-8. We assume that the EPS-
based part is non-blocking (e.g., it uses the \( k \)-ray fat-tree
 topology [31]). Hence, the bandwidth capacity to/from a ToR
switch \( v_s \) through the EPS-based part is only limited by the
ToR switch’s port rate, which is denoted as \( B_{v_s} \). Meanwhile, if
the OXC is configured to bridge the communication between
ToR switches \( v_s \) and \( u_s \), the bandwidth capacity through the
OCS-based part for the switch pair is referred to as \( B_{(v_s,u_s)} \).

The topology of a VNT is modeled as \( G_r(V_r, E_r) \), where \( V_r \)
is the set of VMs and \( E_r \) is the set of SLs that interconnect the
VMs. As the VNT reconfiguration involves VM migrations, we
define the image size of a VM \( v_r \in V_r \) as \( c_{v_r} \). The bandwidth usage of a VL \((v_r, u_r) \in E_r \) is denoted as \( b_{(v_r,u_r)} \).

B. VNT Reconfiguration

We focus on the problem that given the original and new
VNE schemes of several VNTs, how to schedule the actions
for remapping the related VMs and VLs such that the VNT
reconfiguration can be accomplished within the shortest time.

\[
\mathcal{M} = \begin{cases} \mathcal{M}_N : V_r \rightarrow V_s, \\ \mathcal{M}_L : E_r \rightarrow P_s, \end{cases} \quad \mathcal{M}' = \begin{cases} \mathcal{M}'_N : V_r \rightarrow V_s, \\ \mathcal{M}'_L : E_r \rightarrow P_s, \end{cases}
\]

(1)

where \( \mathcal{M} \) and \( \mathcal{M}' \) are the original and new VNE schemes\(^1\) of
a VNT \( G_r(V_r, E_r) \), respectively, \( \mathcal{M}_N \) and \( \mathcal{M}_L \) are the corresponding node and link mapping schemes are \( \{\mathcal{M}_N, \mathcal{M}'_N\} \) and \( \{\mathcal{M}_L, \mathcal{M}'_L\} \), respectively, and \( P_s \) denotes the set of pre-calculated substrate
paths in \( G(V_s, E_s) \). The VNT reconfiguration involves the
rearrangements of related VMs and VLs (i.e., \( \mathcal{M}_N \rightarrow \mathcal{M}'_N \) and \( \mathcal{M}_L \rightarrow \mathcal{M}'_L \)), among which the VL remappings are
performed when the related VMs have been migrated success-
fully, and because they take much shorter time than the VM
migrations, we ignore their latencies. Therefore, our problem is
reduced to how to schedule parallel VM migrations in batches
to realize the VNT reconfiguration within the shortest time.

By analyzing the VM migrations of all the related VNTs,
we can group the VMs whose source and destination racks are
the same as one migration unit (MU), namely, \( m \in \mathcal{M} \), where
\( \mathcal{M} \) is set of all the obtained MUs. As shown in Fig. 1, the
migration of an MU can leverage the bandwidth capacities in

\(^1\)Note that, \( \mathcal{M} \) and \( \mathcal{M}' \) are the preset inputs to the algorithm developed in this work, and thus we will not discuss how to calculate them but concentrate
on the procedure for realizing the VNT reconfiguration based on them.
both the EPS- and OCS-based parts of the HOE-DCN. More illustratively, we use the example in Fig. 2 to explain the VM migrations in an HOE-DCN. Here, we need to migrate three MUs, and based on the current OXC configuration, MUs 1 and 2 can use the bandwidth capacities of both the EPS- and OCS-based parts, while MU 3 can only be migrated through the EPS-based part. Hence, in order to minimize the overall migration latency, we need to schedule the MUs’ migrations in parallel and allocate bandwidth resources in the HOE-DCN to them accordingly, which can be achieved with the algorithms designed in the next section.

**III. ALGORITHM DESIGNS**

In this section, we design two algorithms to realize parallel VM migrations for the VNT reconfiguration in an HOE-DCN, based on the one-shot and multi-shot approaches, respectively.

**A. One-Shot Approach**

In the one-shot approach, we migrate all the VMs in parallel based on the current configuration of the HOE-DCN, and then reconfigure the OXC and related VLs. Then, the problem becomes how to allocate bandwidth to the VM migrations such that their overall migration time can be minimized. This can be described with the following linear programming (LP) model.

**Notations:**
- $M$: the set of MUs.
- $G(V_s, E_s)$: the inter-rack topology of the HOE-DCN.
- $R_v$: the set of rack pairs in the HOE-DCN.
- $b_{v_{in}}$, $b_{v_{out}}$: the available bandwidth capacities to/from rack $v_s$ in $V_s$ through the EPS-based inter-rack, respectively.
- $b_{v_{in}}$: the available bandwidth capacity from rack $v_s$ to rack $v_s$ through the OCS-based inter-rack, and it equals 0 if $u_s$ cannot talk with $v_s$ through the OXC.
- $c_m$: the total image size of all the VMs in MU $m$ in $M$.
- $s_m$: the source rack of MU $m$.
- $d_m$: the destination rack of MU $m$.
- $x_m$: the boolean parameter that equals 1 if rack $v_s$ is the source rack of MU $m$ in $M$, and 0 otherwise.
- $y_m$: the real variable that denotes the bandwidth allocated in the OCS-based inter-rack to migrate MU $m$ in $M$.
- $\xi$: the reciprocal of the overall migration time.

**Variables:**
- $x_m$: the real variable that denotes the bandwidth allocated in the EPS-based inter-rack to migrate MU $m$ in $M$.

**Objective:**
As we invoke the VM migrations in parallel, the overall migration time is just the longest migration time of an MU. For each MU, its migration time can be obtained by dividing its total image size with the allocated bandwidth. Hence, to avoid nonlinearity, we set the optimization objective as to maximize the reciprocal of the overall migration time, which is equivalent to minimizing the overall migration time.

$$\text{Maximize } \xi.$$  \hspace{1cm} \text{(2)}

**Constraints:**

1. 2. 3. 4.

**Eqs. (3)-(4) ensure that the bands allocated in the EPS-based inter-rack for MUs migrated from/to rack $v_s$ in $V_s$ do not exceed the available bandwidth capacities, respectively.

$$\sum_{m \in M} x_m \cdot s_m \leq b_{v_{out}}, \forall v_s \in V_s.$$ \hspace{1cm} \text{(3)}

$$\sum_{m \in M} x_m \cdot d_m \leq b_{v_{in}}, \forall v_s \in V_s.$$ \hspace{1cm} \text{(4)}

**Eq. (5) ensures that the bandwidth allocated in the OCS-based inter-rack for migrating MU $m$ from $s_m$ to $d_m$ does not exceed the available bandwidth capacity.**

$$x_m \geq 0, \forall m \in M,$$ \hspace{1cm} \text{(6)}

**Eq. (6) ensures that the allocated bandwidths are non-negative.**

$$\xi \leq \frac{x_m + y_m}{c_m}, \forall m \in M.$$ \hspace{1cm} \text{(7)}

**Fig. 2. Example on parallel VM migrations in an HOE-DCN.**

**III. A**

**B. Multi-Shot Approach**

Although the LP model can provide us the exact solution to schedule VM migrations in parallel, the one-shot approach might not always lead to the shortest migration time due to bandwidth competition. This motivates us to consider a multi-shot approach. Fig. 3 gives an illustrative example on the comparison between one-shot and multi-shot approaches. Here, for the three MUs, their image sizes are 1, 2 and 1 units, respectively, the bandwidth usages of their VMs are 1, 2 and 1 units/time-unit, respectively, and the available bandwidth capacities on their ToR switches are 3, 1 and 3 units/time-unit, respectively. Therefore, when using the one-shot approach, we cannot reduce the overall migration time due to the bandwidth competition on the ToR switch for Rack 2, i.e., the overall migration time will be $\frac{1+2}{3} = 1$ time-units. On the other hand, if we first migrate MUs 1 and 3 in parallel and then handle MU 2, we can get a shorter overall migration time as $\frac{1+\frac{1}{2}}{5} = 2$ time-units. Note that, the bandwidth competition in Fig. 3 can happen when MUs to/from different racks are being migrated.
Fig. 3. Comparison between one-shot and multi-shot VM migrations.

Algorithm 1: Multi-Shot Parallel VM Migrations

1. \( M_p = \emptyset; \)
2. store values of \( \{b_{v,s}, b_{v,t}\} \) and \( \{b_{u,s}, b_{u,t}\} \) in temporary variables \( \{b_{v,s}, b_{v,t}\} \) and \( \{b_{u,s}, b_{u,t}\} \); 
3. for each rack \( u_s \in U_s \) that has VMs to migrate do 
4. \( M_I = \emptyset; \)
5. \( M_U \) on rack \( u_s \) do 
6. get destination rack of \( m \) as \( u_v \); 
7. migrate \( m \) hypothetically to get the resulting bandwidth usages \( b_{v,s}^{m,t} \) and \( b_{v,t}^{m,t} \); 
8. if \( b_{v,s}^{m,t} < b_{v,t}^{m,t} \) then 
9. insert \( m \) in \( M_I \); 
10. end 
11. end 
12. if \( M_I \neq \emptyset \) then 
13. \( m^* = \text{argmin} \left( \frac{m}{c_{v,s}^{m,t}} \right) \); 
14. insert \( m^* \) in \( M_p \) and remove \( m^* \) from \( M_I \); 
15. migrate \( m^* \) hypothetically to update the values of related \( b_{v,s}^{m^*,t} \) and \( b_{v,t}^{m^*,t} \); 
16. end 
17. if \( M_p \neq \emptyset \) then 
18. migrate all the MUs in \( M_p \) in parallel; 
19. reconfigure the OXC and related VNs and update the status of HOE-DCN; 
20. end 
21. else 
22. migrate all the MUs in \( M_I \) in parallel; 
23. end 
24. break; 
25. end 
26. end
inputs, we utilize the one-shot and multi-shot approaches to schedule parallel VM migrations in batches and realize the VNT reconfiguration. To evaluate our algorithms with different volumes of VMs to migrate, we define a selection ratio $\gamma$, which denotes the ratio of the VMs that are selected to migrate to all the in-service VMs, and changes as $\gamma \in [4\%, 25\%]$ in the simulations. In addition to the proposed algorithms, we also consider a sequential VNT reconfiguration algorithm (i.e., the MUs are migrated one by one) as the benchmark. To guarantee sufficient statistical accuracy, we average the results from 5 independent runs to get each data point.

### B. Migration Time

We first obtain the overall reconfiguration time with the three algorithms, and plot the results in Fig. 4. We can see that the reconfiguration time generally increases with the selection ratio $\gamma$. This is well expected because a larger $\gamma$ means that more selected VMs are given to the VNT reconfiguration algorithms for scheduling their migrations. Meanwhile, when comparing the results obtained in the HOE-DCNs with 6-ray and 8-ray fat-tree topologies, we observe that the reconfiguration time in the 8-ray fat-tree is always longer when other parameters are similar. This is because there are more racks in the 8-ray fat-tree and thus more VMs can be accommodated. In other words, with similar $\gamma$, more VMs are selected in the HOE-DCN with 8-ray fat-tree for VNT reconfiguration.

The results in Fig. 4 also indicate that the sequential reconfiguration algorithm always uses the longest time for VM migrations, while the multi-shot approach achieves the best performance on overall reconfiguration time. This observation verifies that our multi-shot approach can schedule parallel VM migrations in batches, such that the bandwidth competition in the EPS-based inter-rack network is effectively relieved. Moreover, we can see that the reconfiguration time from the multi-shot approach increases much slower than that from the one-shot approach, when the selection ratio $\gamma$ increases. This suggests that the multi-shot approach can maintain the overall reconfiguration time well, even when it needs to schedule a larger number of VM migrations. Therefore, the algorithm’s effectiveness gets further confirmed.

### C. Distribution of Bandwidth Allocations

Next, we investigate the average bandwidth allocated to each VM migration. Fig. 5 shows the results. We can see that sequential reconfiguration algorithm allocates the most bandwidth to each VM migration, followed by the multi-shot and one-shot approaches in sequence. The phenomenon can be explained as follows. First of all, both the sequential reconfiguration and multi-shot parallel reconfiguration algorithms schedule the VM migrations in batches, which means that they can give all the available bandwidths to a portion of the VM migrations in each batch. Hence, their average bandwidth allocations are larger than that of the one-shot approach, which has to distribute the available bandwidths to all the VM migrations simultaneously. Secondly, different from the sequential reconfiguration, which only migrates one MU at a time, the multi-shot approach still invokes parallel VM migrations, i.e., in the same batch, the available bandwidths can be shared by the migrations of several MUs. Therefore, the
average bandwidth allocation from the multi-shot approach is smaller than that of the sequential reconfiguration algorithm. Meanwhile, it can be seen that the average bandwidth allocations from both the one-shot and multi-shot approaches can decrease with $\gamma$, while this will not happen for the sequential reconfiguration. This is because when parallel reconfiguration is considered, both the available bandwidth for VM migration and number of VM migrations in a batch can increase with $\gamma$.

V. CONCLUSION

In this paper, we studied the procedure of reconfiguring VNTs in parallel in HOE-DCNs, i.e., given the original and new VNE schemes of several VNTs, how to schedule parallel VM migrations in batches to realize the VNT reconfiguration within the shortest time. We proposed two parallel VM migration algorithms. The first one used the one-shot approach, and we formulated an LP to solve the scheduling problem for it exactly. Then, to overcome the drawbacks due to the bandwidth competition during parallel VM migrations, we proposed the second algorithm by leveraging the multi-shot approach, i.e., invoking multiple batches of parallel VM migrations such that the migration time can be further reduced. The results from extensive simulations verified that both the one-shot and multi-shot approaches provide much shorter migration time than the benchmark using sequential reconfiguration, and the multi-shot approach achieves the fastest VNT reconfiguration.

ACKNOWLEDGMENTS

This work was supported in part by the NSFC projects 61871357, 61771445 and 61701472, Zhejiang Lab Research Fund 2019LE0AB01, CAS Key Project (QYZDY-SSW-JSC003), and SPR Program of CAS (XDC02070300).

REFERENCES