Soft Computing (2018) 22:7797–7810
https://doi.org/10.1007/s00500-018-3463-8

FOCUS

A virtual cluster embedding approach by coordinating virtual network and software-defined network

Yusong Tan[1] · Rongzhen Li[1] · Qingbo Wu[1] · Jianfeng Zhang[1]

Published online: 14 September 2018

Abstract

Virtual cluster, as a fundamental service of cloud computing, is an important delivery model of cloud services. Allocating physical resources for a virtual cluster is known as virtual cluster embedding (VCE), which has a significant impact on the performance. VCE includes the components of virtual machine, virtual switch, virtual link, virtual software-defined network (SDN) controller and cluster system controller. VCE needs to be considered comprehensively to adapt the influencing factor weight proportion and to be flexible for different types of embedding. This paper, based on the topology of the cloud data center network, develops a coordinated VCE approach, called CoVCE. The approach combines virtual network with SDN to form virtual SDN and weakens slightly the constraints of VM placement. The network centrality, correlation property and resource fragmentation are optimized with multiple objectives to receive more requests, increase the throughput and decrease network delay and runtime. The CoVCE method integrates not only the logical topology of virtual machines in the cluster but also the relationship with virtual switch, virtual link and control services. This does not only optimize the placement location of virtual components, but also allows the virtual resources to migrate upon virtual cluster requests. The method further improves the utilization of physical resources and reduces resource fragmentation. According to extensive simulation and emulation experiments and comparison with correlative algorithms, CoVCE effectively reduces network delays, offers a higher embedding efficiency, improves user experiences and, to some extent, also improves the revenue/cost ratio and throughput.

Keywords Virtual cluster · Software-defined network · Virtual cluster embedding · Multi-objective optimization · Network delay

1 Introduction

VCE is a mapping approach from application-oriented virtual logical resources to physical resources. Mapping virtual cluster (VC) to physical cluster usually includes VM placement (Lin et al. 2017), virtual controller embedding, virtual switch embedding and virtual link embedding, which is similar to virtual network embedding (VNE) and needs to map virtual node and virtual link to physical cluster. However, it is very low efficient to apply directly VNE approach to solve the VCE problem. Firstly, there are more virtual components needed to consider for VCE, for example, VCE has its own controller and the controller placement problem should be solved preferentially. Secondly, some additional node resource constraints, e.g., node location, increase the runtime overhead of VCE. Thirdly, VNE takes account of all the nodes and links to be equivalent and the relation of them is usually neglected, but VCE is closely related to virtual topology and the attributes should be fully used to improve the performance. At last, there are different important role of virtual nodes and links in VCE, and the network delay of data plane is not so rigid as to control plane. In addition, the approach of VNE and the previous VCE have not enough

Communicated by B. B. Gupta.

✉ Rongzhen Li
lirongzhen11@nudt.edu.cn

Yusong Tan
ystan@nudt.edu.cn

Qingbo Wu
qingbo.wu@nudt.edu.cn

Jianfeng Zhang
jfzhang@nudt.edu.cn

[1] College of Computer, National University of Defense Technology, Changsha, China

consider flexible and resilient network flow control for the multi-tenants.

This paper studies the influence factors of VCE and proposes multi-objective optimization heuristic approach (Wang et al. 2018; He et al. 2017) to improve the embedding. The nodes or links of VCE should be given different priority to optimize the performance according to the different importance of the virtual machine/switch, VSDN controller and virtual link in the cluster. The previous method of VCE preferentially takes account of the constraints of VM placement. However, our approach just considers compute resources for VM as a resource pool that only has weak constraints. For example, a leaf node in network topology consists of 32 computing nodes which is compute frame in cloud data center. And the method emphatically adopts the network topology centrality, the correlation property and resource fragmentation optimization to improve the acceptance rate, revenue/cost ratio and throughput and reduce the controller-to-switch and end-to-end latency.

A virtual cluster emulation is developed based on the Mininet (Lantz and O'Connor 2015), OpenVirteX (Al-Shabibi et al. 2014) and OpenDaylight (Medved et al. 2014) for the virtual network control and isolation. The major contributions in this paper are summarized as follows:

1. Designing a virtual network cluster model for controlling and isolating network traffic.
2. Proposing a coordinated VCE approach, which can be aware of the virtual cluster network topology and measures the requirements of nodes/links for VC requests to improve the embedding efficiency and network latency.
3. Applying network centrality, correlation property and resource fragmentation optimization, which can associate the relevant embedding factors of the virtual node and virtual link, to improve the embedding efficiency.
4. The feasibility of the approach is evaluated and compared with other similar algorithms by simulation and emulation experiments.

The rest of this paper is organized as follows. In Sect. 2, we discuss the related work. Section 3 describes the basic concepts. Section 4 combines VN and SDN to optimize the VCE approach for multiple objectives. Section 5 provides the optimization strategies for VCE. In Sect. 6, we describe the VCE algorithm in detail. Section 7 evaluates the benefits of the proposed approaches. Section 8 concludes this paper.

2 Related work

It is a very prominent problem sharing the physical resources and controlling network traffic in the cluster that virtualization technology has been highly concerned. Many researchers have pay attention to the study field of VN, VM, VSDN, VC and other related works. It is a cost-effective to build VC and deliver cloud services for multi-tenants, e.g., the virtual Hadoop service and virtual HPC service. VC greatly simplifies the service model and improves the flexibility of the system but, to a large extent, also leads to the problem of VCE. The VCRs must be mapped to the physical cluster, and the embedding needs to satisfy the constraints of nodes and links, which is similar to the VNE problem and also the NP-hard problem (Fischer et al. 2013; Wei et al. 2014).

Some previous studies have proposed the related problems of VCE. Wei et al. (2014) formulated a topology-aware partial virtual cluster mapping algorithm. Yu et al. (2008) developed a path splitting mechanism and multi-commodity flow to optimize VNE problem. Chowdhury et al. (2012) proposed the mixed integer program and relaxed the constraints of VNE through the network aggregation approach. Butt et al. (2010) developed a topology-aware approach which can identify bottlenecks of the physical nodes or links. Wang et al. (2012) proposed network centrality analysis to optimize VNE based on network topology. Pang et al. (2014) described hardware network reliable communication and developed a high-bandwidth and low-latency network. Mehmet and Mostafa (2014) proposed a VNE approach based on SDN which is to balance the load on the substrate network and minimize controller-to-switch delays. Rabbani et al. (2013) presented a virtual data center embedding solution to improve the utilization of physical substrate. Gomes et al. (2014) proposed an architecture to avoid the waste of ISP resources. Zhou et al. (2016) presented a resource allocation algorithm combined channel with power allocation design to realize energy saving. Meanwhile, virtual SDN embedding has been studied extensively. Papagianni et al. (2014) studied the virtual topology embedding approach in the SDN-enabled networked cloud environments. Li et al. (2014) proposed a novel online GraphMap algorithm and adopted a top-k dominating model to rank the nodes to improve resource allocation. Lin et al. (2015) proposed a resource allocation algorithm based on equivalent optimization. Wang et al. (2017) introduced a scheduling algorithm named Recom_Task _Assign to address multi-dimensional loops applications on heterogeneous multicore processors.

Most of the current researches do not deeply takes into consideration the network topology of VCE and the correlation properties of physical nodes. Applying the VNE solutions to VCE problem is not efficient directly, which may affect the costs and benefits. Existing algorithms treat all the nodes or links equally, while the role of node or link is different, such as different workloads, different roles for control nodes and compute nodes, etc. This paper develops a coordinated VCE approach, coordinating VM, virtual switch (vswitch) and virtual controller (vcontroller), which combines the network topology with link splitting features, correlation property and resource fragmentation optimization, and optimizes the CoVCE algorithm which effectively improves the performance and network delay.

3 Virtual cluster modeling and problem description

3.1 Virtual network cluster model

The previous approach of VCE mainly refers to VM mapping and VN embedding; however, it is hard to timely control the network traffic of virtual cluster. Meanwhile, there are some shortcomings in the network traffic control to the network delay and the limits of bandwidth and the relationship of the topology attributes is slightly neglected. In order to solve these problems, this paper proposes a coordinated virtual network cluster model combined VN with SDN, which is a customized interconnecting cluster composed of a certain number of virtual components that includes VCU, vswitch, vcontroller, controller-to-switch link and virtual link.

The coordinated approach aims to improve network traffic, throughout, and the performance of the cluster. The virtual cluster embedding model is shown in Fig. 1, which demonstrates virtual cluster with controller mapping into the physical infrastructure. Firstly, a complete virtualized SDN environment is built, which is based on a virtual network, and a network environment with different network protocols and software-defined network can be provided to different tenants. Secondly, the physical network infrastructure to implement virtual network may be based on a general-purpose ICT device or an SDN physical network infrastructure. Therefore, it can be seen that the virtualized SDN is not strongly related to the hardware infrastructure. The coordination of virtual software-defined network (VSDN), virtual networks (VN) and virtual machine (VM) is aim to improve the traffic control, quality of service, the flexibility and manageability of cluster system. The components of virtual cluster are shown in Fig. 2, and the conception of virtual cluster is shown in Table 1.

In order to optimize these existing problems including the similar (Sun et al. 2018; Zhang et al. 2017; Jiang et al. 2018), VCE in this paper is defined to a multi-objective optimization problem that needs to be studied in-depth in cloud data center, which plays a critical role in resource utilization and reliability of the cloud data center as well as the users' experiences (Yu et al. 2018; Wang et al. 2018). This paper models a coordinated virtual network cluster and then launches and explores gradually the research on the approach of virtual cluster embedding based on the topology. On the other hand, it is seen that VM is interconnected with virtualized SDN, which is organized to build VNC. VNC is a new abstract conception of cloud services, which has a variety of conceptual descriptions from different perspectives. In this paper, the conception of virtual cluster takes into account the network communication delay and network propagation delay as an important metrics based on virtual network topology, using the topology-aware method to measure the core nodes and multi-path technology to enhance the virtual network cluster survivability. The formal description is as follows:

Table 1 Concept description of coordinated virtual cluster embedding

Virtual node	VM and virtual switch
Controller node	Virtual cluster system controller and VSDN controller
Logical path	Virtual link and controller-to-switch link

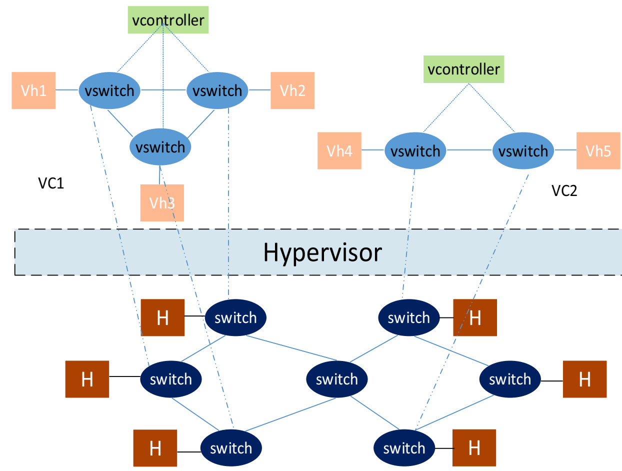

Fig. 1 Coordinated virtual cluster embedding model

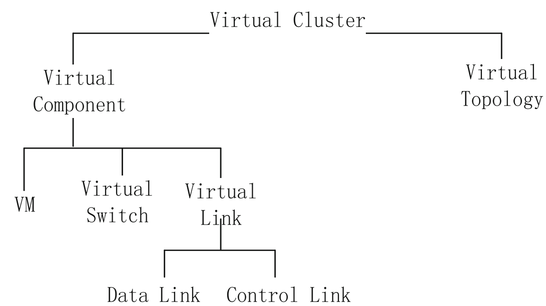

Fig. 2 Components of virtual network cluster

Definition 1 (*virtual network cluster, VNC*) Virtual network cluster is composed of virtual cluster unit that is interconnected by the vswitch and controlled with VSDN controller, which is the basic unit to deliver cloud tenant services. The virtual network cluster can be represented by VNC (n, b, dl), where n is the number of virtual cluster units (instances), b is the network bandwidth, and dl is the network delay.

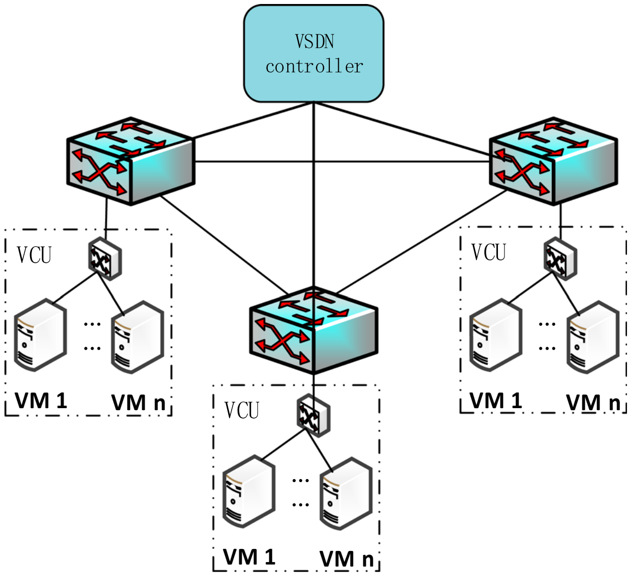

Fig. 3 Virtual network cluster architecture

VNC typically refers to cloud tenant request that is used to represent resource requirement. In the production system, it will change with time to the user's predefined resource request so that the service provider should dynamically adjust the resources according to the actual demand. In order to no against SLA and secure problems (Li et al. 2014, 2018; Zhu et al. 2018; Cai et al. 2018; Jhaveri et al. 2018; Mijumbi et al. 2014; Zhang et al. 2018), network traffic should be effectively controlled in the SDN network architecture by the separation of control plane and data plane. The virtual network cluster architecture is shown in Fig. 3.

Definition 2 (*virtual cluster unit, VCU*) Virtual cluster unit is the basic unit of the virtual network cluster model that virtual switch adopts the hose model to connect VM. VCU can be represented by VCU (c, m, b, d), where c is the computing power, m is the memory, b is the network bandwidth, and d is the disk capacity.

In large-scale cloud data center, however, the network design often shrinks the bandwidth to support the scale and save the expenditure, which leads to the communication among nodes through different access switches that is difficult to protect all links 1:1 bandwidth among VMs. In order to optimize the network bandwidth, the traffic reflection of the virtual cluster unit (VCU) is beneficial to maximize the probability of 1:1 network bandwidth among VMs. The main feature of the virtual cluster unit is that the traffic belonging to the same virtual cluster unit can be reflected back through the virtual switch. Analogous to cloud resource pool, virtual cluster unit is a local resource pool and also known as an instance of the virtual network cluster. VNC can allocate resources according to the globally visible list of active

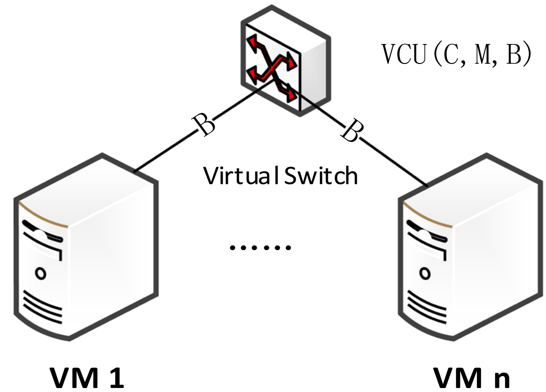

Fig. 4 Virtual cluster unit

resources which is provided positively by the virtual cluster unit. To protect the data's trust worth and user privacy in the cloud, some methods are proposed (Zhu et al. 2017; Li et al. 2014; Lin et al. 2017; Guan et al. 2017, 2018; Tan et al. 2018). Meanwhile, in cloud data center, the virtual cluster unit is embedding the resource pool that is composed of the nearby physical nodes, located in a compute frames or compute rack, through virtualization technology, such as KVM, Docker, that satisfies the minimum network latency. The virtual cluster unit is shown in Fig. 4.

Definition 3 (*physical cloud data center, PCDC*) Physical cloud data center is modeled as a weighted undirected graph $Gp=(Np, Lp, An, Al)$, where Np and Lp are, respectively, the set of physical nodes and links. Let C refer to the set of controller placement in physical SDN and lc denote the total number of control paths which are selected by topology-aware shortest path method to reduce propagation delay (hops). An and Al are defined as the constraints of physical nodes and links, respectively. For example, $An(c)$ refers to the CPU capability of physical node and $Al(b)$ represents the bandwidth of physical link.

Definition 4 (*virtual cluster request, VCR*) Virtual cluster request which is distinct from physical infrastructure is a logical topology of the tenants' requirements. It refers to the vertex and edge of the graph and also models as a weighted undirected graph $Gv=(Vv, Ev, Rv, Re)$, where Vv and Ev, respectively, represent with the set of virtual nodes and virtual links. Rv and Re are defined to the requirements of virtual node and virtual links. For instance, $Rv(c)$ refers to the CPU capability of virtual node and $Re(b)$ represents the bandwidth of virtual link.

Definition 5 (*virtual cluster embedding, VCE*) Virtual cluster embedding is usually defined to the constrained resources allocation that Gv embeds the subgraph of Gp that should meet the resource constraints of Rv and Re for the virtual cluster request of cloud tenants. $M: Gv \rightarrow (N', P', Rn, Rl)$,

Table 2 Notations of VCE

Gp	Physical cloud data center
Np	Nodes of physical cloud data center
Lp	Links of physical cloud data center
An	Node attribute of physical cloud data center
Al	Link attribute of physical cloud data center
Ps	Path on physical cloud data center
Gv	Virtual cluster
Vv	Nodes of virtual cluster
Ev	Links of virtual cluster
Rv	Node requirements of physical cloud data center
Re	Link requirements of physical cloud data center
Rn	Physical node resources allocated to virtual node
Rl	Physical link resources allocated to virtual link
fn	Function of mapping virtual node to physical node
fl	Function of mapping virtual link to physical path

N' contained in *Np*, *P'* contained in *Ps* and *Ps* is set of *Lp*. *Rn* and *Rl* are defined to the physical resource allocation for virtual node and virtual links, respectively. The process of VCE is shown in Fig. 1.

Mapping can be divided into two stages:

Node mapping: which includes the mapping of virtual controller and virtual switch, is defined as: $fn: Vv \rightarrow Np$, mapping the virtual nodes to physical nodes where the arbitrary v_v belongs to *Vv* and existing n_s belongs to *Np*, $n_s=fn(v_v)$ and $An(c) \geq Rv(c)$.

Link mapping: which includes the mapping of virtual link and controller-to-switch link, is defined as: $fl: Ev \rightarrow Ps$, mapping virtual links to physical path where the arbitrary e_v belongs to *Ev* and existing p_s belongs to the *Ps*, $p_s=fl(e_v)$ and $An(b) \geq Re(b)$, where *Ps* consists of a subset of the physical path set *Lp*, p_s consists of a subset of the physical path *Lp*, and the concept is shown in Table 2.

3.2 Objectives

This paper takes into account multi-factor multi-objective optimization approach to solve the problem and adopts the network centrality, correlation property and resource fragmentation to design and optimize the proposed approach. VCE coordinated VN and SDN have a new problem that is needed to study extensively to provide the best cloud services, which should map VCRs to physical cluster as many as possible. The objective is to maximize mapping success rate, revenue/cost ratio and throughput and to minimize the embedding runtime, resource consumption and network delay. Similar to the previous works (Wei et al. 2014; Yu et al. 2008; Tan et al. 2018), we define its revenue $R(Gv(t))$ at any particular time t that the virtual cluster *Gv* is running as:

$$R(Gv(t)) = \sum_{e_v \in Ev} \mathrm{BW}(e_v) + \sum_{v_v \in Vv} \mathrm{CPU}(v_v) \tag{1}$$

where $\mathrm{BW}(e_v)$ and $\mathrm{CPU}(v_v)$, respectively, represent the bandwidth of E_v and CPU requirements of V_v. Similar to Medved et al. (2014), the cost of a request *Gv* is defined as the sum of resources allocated to the virtual cluster requests.

$$C(Gv(t)) = \sum_{e_v \in Ev} \mathrm{BW}(e_v) * \mathrm{length}(e_v) + \sum_{v_v \in Vv} \mathrm{CPU}(v_v) \tag{2}$$

where length (e_v), similar to the number of network hop, is the length of the loop-free path and the greater the value, the more the link bandwidth is consumed.

The revenue-to-cost ratio (*R/C*) is used to measure the quality of the virtual cluster embedding, which is defined as:

$$R/C = R(Gv(t))/C(Gv(t)) \tag{3}$$

According to (1) and (2), where the larger value means to be better embedding quality, the range of *R/C* value is between 0 and 1. The runtime of virtual cluster embedding represents with the total embedding time of virtual nodes and virtual links.

$$sumT = \sum_{e_v \in Ev} T(e_v) + \sum_{v_v \in Vv} T(v_v) \tag{4}$$

where $T(e_v)$ and $T(v_v)$ refer to the consumed time by virtual links and virtual nodes embedding, respectively. To achieve long-term success of the virtual cluster request acceptance rate, the acceptance rate is defined as:

$$\mathrm{AR} = \lim \frac{\sum_{t=0}^{T} \mathrm{VCR}_s}{\sum_{t=0}^{T} \mathrm{VCR}} \tag{5}$$

VCR_s represents the number of virtual cluster requests successfully mapped and VCR represents the total number of virtual cluster requests.

The average network delay of the controller-to-switch is defined as:

$$\text{Delay} = \frac{\sum_{l_{cv} \in Lc} D(l_{cv})}{|L_c|} \quad (6)$$

Lc is the set of controller-to-switch, $|L_c|$ denotes the number of Lc, and $D(l_{cv})$ is the delay of each controller-to-switch.

The approach, which balances multi-objective embedding factors, should be explored sufficiently to maximize the *AR*, *R/C* and throughput by successfully mapping more virtual cluster requests into physical cloud data center and reduce the virtual cluster embedding runtime and network latency.

4 Multi-objective optimization coordinated VN and SDN

Mapping VC with VSDN needs to meet the following constraints: (1) satisfying the resource constraints of the vswitch, VSDN controller, controller-to-switch and virtual link; (2) minimizing the cost of VCE; (3) minimizing the average controller-to-switch latency and end-to-end latency; (4) maximizing the embedding success rate, *R/C* and *AR*; (5) maximizing throughput.

This paper defines the binary variable set X_{uv} as the request variable for the embedding from virtual node u into physical node v.

$$x_{uv} = \begin{cases} 1 \text{ if } u \text{ is embedding into } v \\ 0 \text{ otherwise} \end{cases} \quad (7)$$

A binary variable for virtual link uu' to physical link vv' embedding is represented by $y_{(uu')(vv')}$.

$$y_{(uu')(vv')} = \begin{cases} 1 \text{ if } uu' \text{ is embedding into } vv' \\ 0 \text{ otherwise} \end{cases} \quad (8)$$

The VCE, firstly, should satisfy the following constraints.

$$\sum_{v \in N_{\mathrm{p}}^s + N_{\mathrm{p}}^x} x_{uv} = 1, \quad \forall u \in V_{\mathrm{v}}^s \cup V_{\mathrm{v}}^x \quad (9)$$

$$\sum_{u \in V_{\mathrm{v}}^s \cup V_{\mathrm{v}}^x} x_{uv} \times Rv_i(u) \leq An_i(v), \quad \forall v \in N_{\mathrm{p}}^s \cup N_{\mathrm{p}}^x, \forall i \quad (10)$$

where N_{p}^s refers to the physical switch nodes and N_{p}^x refers to the physical server nodes, V_{v}^s refers to the virtual switch nodes, and V_{v}^x refers to VMs. Constraint (9) ensures that one virtual node is just only mapping into a single physical node and each virtual node must successfully map into physical node. Constraint (10) guarantees that the infrastructure capacities cannot be less than the resource requirements of the VCRs.

Similarly, virtual link mapping should satisfy the following restrictions.

$$\sum_{vv' \in Lp} y_{(uu')(vv')} \geq 1, \quad \forall uu' \in Ev \quad (11)$$

$$\sum_{uu' \in Ev} y_{(uu')(vv')} \times Re(uu') \leq Al(vv'), \forall vv' \in Lp \quad (12)$$

$$x_{uv} = 1, xu'v' = 1, \quad \forall u, u' \in V_{\mathrm{v}}^s \cup V_{\mathrm{v}}^x \quad (13)$$

Constraint (11) ensures that a single virtual link is mapped to a physical path. Constraint (12) guarantees that the physical link has sufficient resources to accommodate the virtual link. Constraint (13) means that virtual link mapping requires a virtual link to be successfully mapped to a physical path and a mapped virtual link belongs to the link that locates between the successfully mapped nodes. Next, we study the VCE problem from three aspects: virtual controller/switch embedding, virtual link embedding and VCU placement.

4.1 Virtual controller/switch embedding

Virtual node embedding needs to select the appropriate physical nodes to satisfy the resource constraints of the controller or vswitch requests of the VCR. Given the characteristics of the vswitch to simplify embedding parameters, TCAM is the important restrictive resources for virtual node embedding, which should satisfy the basic requirements as (14). In order to simplify the process of virtual node embedding, TCAM is considered only as the constraint parameter and the limitation of CPU is similar to TCAM, and it can be expressed by function *fn*.

$$\sum_{u \in V_{\mathrm{v}}^s} x_{uv} \times \mathrm{TCAM}_{vi}(u) \leq \mathrm{TCAM}_{ni}(v), \quad \forall v \in N_{\mathrm{p}}^s, \forall i \quad (14)$$

where $\mathrm{TCAM}_{vi}(u)$ refers to the requirement of virtual node u and $\mathrm{TCAM}_{ni}(v)$ refers to the residual resources of physical node v.

VSDN controller placement is a special kind of vswitch embedding, which is related to the left virtual nodes mapping and has an influence on the performance of VCE. It should give the priority to VSDN controller mapping to optimize embedding approach and controller-to-switch link selection to obtain the minimum network delay.

4.2 Virtual link embedding

Virtual link embedding can be divided into two methods: the unsplittable shortest path and the splitting multi-path. Considering the network delay and the mapping efficiency, this paper maps the controller-to-switch link through the unsplittable shortest path and other link embedding adopts multi-path method to enhance the embedding success rate. The k-shortest algorithm is commonly used by shortest path method, while path splitting has a variety of splitting strategies that can effectively manage available bandwidth to achieve better resources utilization and allows the physical cluster to accept more VCRs. For example, multi-commodity flow method can effectively optimize the mapping performance. A group of virtual links produces a set of r commodities, and the method should satisfy the basic requirements as (15).

$$\sum_{i=1}^{r} y_{(uu')(vv')} \times fl(ci, uu') \leq \mathrm{BW}(vv'), \quad \forall vv' \in Lp, \quad uu' \in Ev \tag{15}$$

where $fl(c_i, uu')$ is the bandwidth of virtual link uu' that allocate to commodity c_i to the corresponding virtual link. For the multi-path method, it is necessary to ensure that the requested bandwidth of any two virtual nodes n_i and n_j is no more than the sum of all the available selected paths in order to satisfy the bandwidth limitation.

4.3 VCU mapping

VCU is the fundamental unit that actually performs the calculation, and its resources are just only constraining the total amount of the resources for the VCRs. The goal is to map each instance to the physical servers as soon as possible, and the mapping should satisfy the resource requests in the cluster. A lot of strategies can be used to map VCU to physical servers, but it needs to meet the following basic restrictions as (16) for computing resources and other attributes is same to it.

$$\sum_{u \in V_{\mathrm{v}}^{s}} x_{uv} \times \mathrm{CPU}vi(u) \leq \mathrm{CPU}ni(v), \quad \forall v \in N_{\mathrm{p}}^{s}, \forall i \tag{16}$$

4.4 Overall formulation of virtual cluster embedding

The main objectives of this paper can be summarized as minimizing resource costs and network latency, and the main strategy is to adjust the different importance of influence factor by changing the proportion of factors as formulation (17) to implement different targets. Minimizing resource overhead can potentially increase resource utilization and the success rate of the embedding, and minimizing the network latency can improve system throughput with sufficient computing resources. Therefore, with the aforementioned respective formulations for the VCE, we formulate the VCE problem into a multi-objective optimization.

$$\mathrm{Min}\left(\alpha \sum_{e_v \in Ev} \mathrm{BW}(e_v) * \mathrm{length}(e_v) + \beta \sum_{v_v \in Vv} \mathrm{CPU}(v_v) + \chi \mathrm{Delay}\right) \tag{17}$$

where α, β and χ are the adjustable influence factors for the multi-objective optimization.

The capacity constraints of the multi-objective optimization are shown in Eqs. (18), (19), the TCAM and the CPU must not exceed the upper limit of the physical resources, and the virtual link bandwidth must not exceed the bandwidth constraints of the physical link.

$$\begin{cases} \sum_{u \in V_{\mathrm{v}}^{s}} x_{uv} \times \mathrm{TCAM}_{vi}(u) \leq \mathrm{TCAM}_{ni}(v), \quad \forall v \in N_{\mathrm{p}}^{s}, \forall i \\ \sum_{u \in V_{\mathrm{v}}^{s}} x_{uv} \times \mathrm{CPU}_{vi}(u) \leq \mathrm{CPU}_{ni}(v), \quad \forall v \in N_{\mathrm{p}}^{s}, \forall i \end{cases} \tag{18}$$

$$\sum_{i=1}^{r} y_{(uu')(vv')} \times fl(ci, uu') + \sum_{i=1}^{r} y_{(uu')(vv')} \times fl(lc, uu') \leq \mathrm{BW}(vv'), \forall vv' \in Lp, uu' \in Ev \tag{19}$$

5 Optimizing strategy of virtual cluster embeddinG

5.1 Network centrality

Network centrality is an important metrics for the network topology. Degree, closeness, betweenness and eigenvector (Gomes et al. 2016) are commonly used to measure the centrality of network topology. However, the existing related works do not sufficiently take into consideration of network topology, although some of the existing works (Salvadori et al. 2011; Li et al. 2016; Yu et al. 2018) refer to the attributes of network topology and justly use capacity and bandwidth to assess the importance of node or link. For taking full advantage of the topological properties and improving the acceptance rate and increasing revenue, we make use of the relation of the links and nodes with other adjacent to estimate the importance of the node/link and apply multi-path to reduce the network bandwidth fragmentation and optimize the utilization of network bandwidth. The degree of a node is the number of immediate adjacent links to its neighbors that can be explained by the immediate possibility that a node interconnects closely with others, the formulation of degree is defined as:

$$D(n_i) = \deg(n_i) \tag{20}$$

$\deg(n_i)$ refers to the number of link which is adjacent to n_i. Closeness, the two direct adjacent points, represents that the two points are arbitrarily near to each other in any case and is a sophisticated measure of centrality. The closeness of a node n_i refers to the sum of geodesic distances to all other reachable nodes. The formulation of closeness can be denoted as our proposed work (Li et al. 2016):

$$Cc(ni) = 1/\sum_{j=1}^{n} d(ni, nj) \tag{21}$$

$d(n_i, n_j)$ refers to the distance of the shortest path between nodes n_i and n_j. Whatever n_i is close to the central location of the network topology and its closeness value will be relatively large, or else the closeness of n_i is in boundary position that has a small value. Betweenness quantifies the number of times a node acts as a bridge, and the betweenness of a node is the number of shortest paths that all nodes can reach at all other nodes through the node, thus named visually this node as a bridge node. For VCRs, the bridge node can carry more node flows, and it means more optimization location choices and available resources by giving preference to be mapped into physical nodes (He et al. 2016; Chen et al. 2014). The betweenness of node n_i id defined as:

$$B(n_i) = \frac{s_i}{S} \tag{22}$$

$B(n_i)$ represents the betweenness of node n_i, s_i denotes the number of shortest path through node n_i, and S is the number of shortest path of all nodes.

5.2 Correlation property

VCE usually only considers the attributes of the unmapped nodes. The relations between the mapped and unmapped nodes are usually ignored. It may lead to require multiple hops to support the logical topology and be higher network latency.

In order to overcome this problem, we adopt the relations between the mapped nodes and the unmapped nodes when sorting the physical nodes, and the relationship is defined as follows:

$$\mathrm{CoP}(n_i) = \lambda^{\sum_{j=1}^{k} d(i,j)/\min(bw(i,j))} \tag{23}$$

where $\mathrm{CoP}(n_i)$ indicates the correlation between the unselected physical node n_i and all the selected nodes, λ is a Euler' constant, $d(i, j)$ is the distance of node i and node j, $\min(bw(i, j)$ refers to the minimum bandwidth of node i and node j, the correlation is shown in Fig. 5.

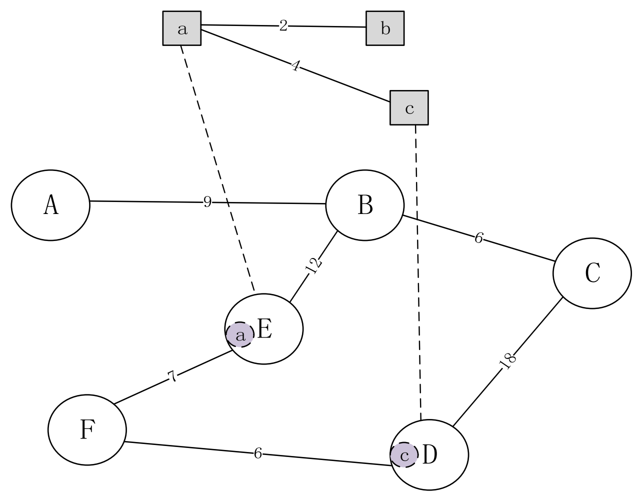

Fig. 5 An example of VCE correlation property

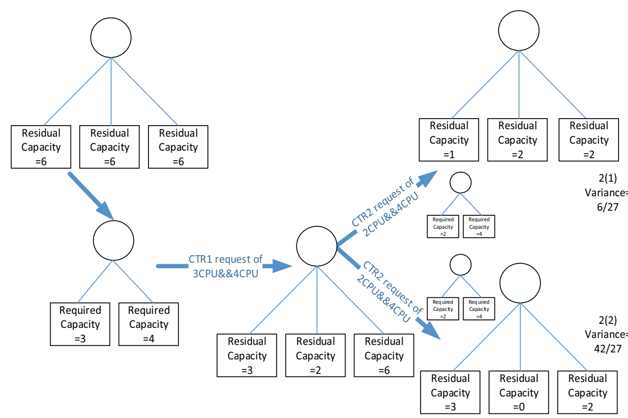

Fig. 6 An instance of resource fragmentation optimization

For example, the correlation property $\mathrm{CoP}(B)$ of B is determined by $\min(d/\mathrm{BW})$, which is better than other node weights for the next virtual node mapping.

5.3 Resource fragmentation

Resource fragmentation is another important factor. For a given network topology, the remaining resources of the nodes reflect whether the physical cluster can accept new VCRs.

For example, after accepting a two nodes VC1 request with three CPUs and four CPUs, the remaining physical cluster CPUs are 3, 2 and 6, respectively, as shown in Fig. 6. It assumes that the next two requests VC2 and VC3 are 2, 4 and 2, 3, respectively, for the requirements of CPU resources. Then, the VC3 can be accepted with large visible variance and the smaller variance cannot accept VC3 due to resource fragmentation. It can be seen that the larger the variance is, the fewer resource fragment is, and the more the VCRs can be accepted.

According to our research, the fragmentation factor that affects the VCE is defined as:

$$\gamma_{cpu} = \frac{\sum\limits_{n_i \in Np} \left(r_{c(ni)} - \overline{r_c}\right)^2}{|Np|} \tag{24}$$

$rc(ni)$ is defined to the residual resources, and $\overline{r_i}$ refers to the average remaining capacity of the physical resources, for instance, $\overline{r_c}$ denotes the value of the remaining resources CPU, $|Np|$ indicates the number of nodes in the physical cluster.

5.4 Combinatorial optimization

The closer the node is, the smaller the number of the hop is, the less the network delay is, the formula is shown as (25). The switching capability is very important to the network traffic in VC. Therefore, our method adopts the betweenness property of the network centrality when the vswitch is mapped; the formula is shown as (26). There are some differences between the nodes of physical cluster and nodes of virtual cluster. For example, it has the related property between the node in network topology of the physical cluster that has accepted the virtual node and the node that is not selected by the VCR, which will affect the performance of VCE. In addition, resource fragmentation of the physical node also will affect the embedding of virtual cluster, so we need to consider these comprehensive factors, and the formula is shown as (27).

$$R_{\mathrm{dc}}(n_i) = \mathrm{TCAM}(n_i)\mathrm{D}(n_i)[C_c(n_i)]^2 \left[\sum_{l \in L(n_i)} \mathrm{BW}(l)\right] \tag{25}$$

$$R_{\mathrm{dcb}}(n_i) = \mathrm{TCAM}(n_i)\mathrm{D}(n_i)[C_c(n_i)]^2 B(bi) \left[\sum_{l \in L(n_i)} \mathrm{BW}(l)\right] \tag{26}$$

$$R_{\mathrm{cop}}(n_i) = (\gamma_{\mathrm{cpu}})\mathrm{COP}(n_i)\mathrm{TCAM}(n_i)\mathrm{D}(n_i)[C_c(n_i)]^2 \times \left[\sum_{l \in L(n_i)} \mathrm{BW}(l)\right] \tag{27}$$

By studying the different influence factors, VSDN controller, vswitch and physical node resources are optimized, respectively.

6 VCE heuristic algorithm with VSDN controller

According to our previous research, VCE can be divided into four parts: controller embedding, vswitch embedding, virtual link embedding and VCU placement. So that computational complexity is also the polynomial time. The algorithm preferentially maps the VSDN controller of the VCR to the physical nodes, and then, the algorithm maps the vswitch of the VCR to the physical nodes. If the VSDN controller and vswitch is all successfully mapped, the node edge will be mapped to the physical path according to the location of the virtual nodes. VCE must take into fully account the network topology of the virtual cluster and synthetically evaluates the closeness, degree and betweenness of the network centrality, the correlation property and the resource fragmentation to improve the performance. Finally, VCU is mapped to physical servers according the list tables that proactively send to VCE, which has been described in section III.

6.1 VSDN controller placement algorithm

According to the formulation (25), the mapping of VSDN controller takes fully use of the network topology closeness attribute of the network centrality for VSDN controller placement. The process is shown in Algorithm 1.

Algorithm 1 VSDN Controller Placement Algorithm

```
Input:
Gp: current physical cloud data center
Gvi the ith arriving VC request
VSDN controller request
/*VSDN controller placement solution*/
Calculate Rcd (ni) of all residual physical nodes using(25) in noincreasing order
Rank physical nodes according to the calculating results
for all the left virtual nodes do
  Mapping VSDN controller of Gvi to the physical node with largest rank that
   satisfied every constraints and the node is unmapped with any other nodes
   from the VCRs
  if failed to map VSDN controller vi then
      return FAILURE
  end if
end for
return CTR_MAPPING_SUCCESS
```

6.2 Virtual switch embedding algorithm

Algorithm 2 Vswitch Embedding Algorithm

```
Input:
Gp: current physical cloud data center
Gvi the ith arriving VC request
/*mapping vswitch solution*/
Calculate Rcdp(ni) of all residual physical nodes using(27) in
noincreasing order
Calculate Rcdb(ni) of all constrained virtual nodes using(26) in
noincreasing order
Rank physical nodes and virtual nodes according to the calculating results
for all the unembedded virtual nodes do
   Choose the virtual node ni with largest values
   Mapping vi to the physical cluster that
     a. Unmapped
     b. Meet the requirement of vi
     c. With the largest value
  if failed to map vi then
  Return FAILURE
   end if
end for
return NODE_MAPPING_SUCCES
```

Physical nodes are sorted in no-increasing order according to the formulation (27). When VCR arrives, virtual nodes are sorted in no-increasing order by formulation (26). The virtual node with the largest weight is mapped to the node with the largest weight. If the CPU and bandwidth of the node satisfy the VCR, the virtual node is mapped successfully. Then, this process will be repeated for all left unmapped virtual nodes. If there are any virtual nodes that do not satisfy the constraints of the request, the VCR will be placed in the queue. The VCR is mapped successfully only if all virtual nodes are mapped successfully in the time window. Otherwise, this VCR will be discarded. The process is shown in Algorithm 2.

6.3 Virtual link embedding algorithm

Once virtual nodes are mapped successfully, it tries to map virtual links. The algorithm will choose the different virtual link embedding strategies for VSDN controller and vswitch embedding according to the return value. If the return value is CTR_MAPPING_SUCCESS for VSDN controller, the k-shortest method is used. Otherwise, multi-path method is used; the specific process is shown in Algorithm 3.

Algorithm 3 Virtual Link Embedding Algorithm

```
Input:
Ev: The set of virtual links
Lp: The set of physical links
/*mapping virtual link into physical path*/
for all requests that node mapped successfully do
  if controller-to-switch then
     Search the kshortest path between controller and the selected nodes
     if find one physical path with adequate bandwidth
       Assign the virtual link to this path
  else if using equation (15) generates r commodities then
     multicommodity_flow(ev(i, j) )
     return 0
  else
     Find the bottleneck substrate link
     Randomly choose one virtual link that mapped at the bot tleneck link
     mapping one virtual node of the virtual link into another substrate node
       with maximum redusial resource Rdc defined qeuations(7)
   (multicommodity_flow(ev(i, j))
    end if
  end if
end for
```

6.4 CoVCE embedding algorithm

Algorithm 4 CoVCE Embedding Algorithm

```
Input:
Gp: current physical cloud data center
Gvi the ith arriving VC request
/*mapping virtual cluster into physical cluster*/
Sort the VCRs according to their revenues
if no VCR left, stop
  for all unmapped VCRs do
    Take one VCR with the largest revenue
  Map VSDN Controller and vswitch of Gvi using algo rithm(1) and (2)
  if CTR_MAPPING_SUCCESS&& NODE_MAPPING_SUCCESS then
      Mapping the virtual links of the request using virtual link embedding
           Algorithm (3)
  else if  LINK_MAPPING_SUCCESS then
     Occupy the physical resources and update the state of physical cluster
     Set the state of this Gvi request to EMBED DING_SUCCESS
     continue
  else
     Set the state of this request to EMBEDDING_FAILED
  Mapping VM to ph ysical server randomly
    end if
  end if
 end for
end if
```

VCR is sorted in no-increasing order according to the revenue. The approach sorts the requests in every constant time interval, which is a feasible for batch or real-time tasks. Once getting the largest VCR, the embedding of VSDN controller and vswitch is executed on the basis of the above Algorithms (1) and (2), respectively; then, it will decide whether to embed virtual link based on the results of VSDN controller and vswitch embedding. And only if both the VSDN controller and the vswitch are embedded successfully, the virtual link mappings are actually required. At last, VCU is mapped into physical servers using random method, CoVCE is described in detail in algorithm 4.

7 Experimental evaluation

We explore the implementation of product environment (Sun et al. 2018; Yu et al. 2018) and develop BL, IC, TMVCE and CoVCE algorithms based on the experimental environment and compare the runtime overhead, network delay, acceptance ratio, long-term R/C and throughput by the experimental data analysis. The results show CoVCE algorithm based on network centrality, correlation property and resource fragmentation optimization has certain advantages in the R/C ratio, network delay and throughput of virtual cluster requests.

7.1 Simulation

The simulation environment is described in our previous work TMVCE (Li et al. 2016) and adds the network bandwidth delay variables to verify the latency of virtual cluster. The physical link delay of physical cloud data center network follows uniform distribution from 1 ms to 20 ms, and virtual links are real numbers uniformly distributed between 1 and 50 ms. Suppose that there is only one VSDN controller per virtual cluster and each virtual switch can connect VCUs with enough resources. This paper compares BL, IC and TMVCE and the proposed CoVCE algorithm where the notations used to refer to different algorithms are presented in Table 3. In order to fairly compare the similar algorithms, some weak restriction attributes of virtual cluster topology, such as tolerating resource downgrade of the virtual cluster request, unrestricted position of VMs, are selected to evaluate them. A large number of the tests for virtual cluster embedding have been done for the actual effect through the experimental data analysis. Firstly, the runtime of virtual cluster embedding has a significant impact on cloud tenants' experience. It is effective to improve the efficiency of virtual cluster embedding and reduce the embedding runtime for improving the experience.

Experimental results show that CoVCE consumes less runtime than BL algorithm and approximates the runtime compared to IC and TM, which is shown in Fig. 7. The main reason is to optimize the virtual cluster nodes and link mapping by degree, closeness and betweenness, and correlation property and resource fragmentation as shown in Eq. (27), while the proposed CoVCE approach increases the restrictive factor of the location selection of the VSDN controller to reduce network delay.

Table 3 Algorithm representation

Notation	Description
BL	Baseline algorithm including node/edge mapping (Yu et al. 2008)
IC	Node ranking based on improved closeness centrality considering both topology and the resources (Wang et al. 2012)
TMVCE	Topology-aware and multi-path virtual cluster embedding algorithm (Salvadori et al. 2011)
CoVCE	Coordinated VC embedding heuristic algorithm with VSDN controller [our proposed]

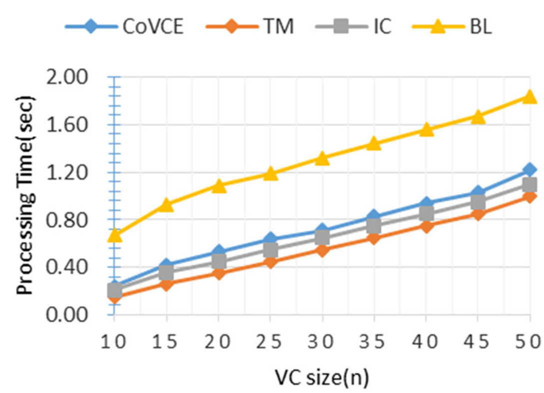

Fig. 7 Comparison of four algorithm processing time

It is shown in Fig. 8 the acceptance rate of the virtual cluster embedding is further improved by the network centrality and correlation property optimization. Compared with the BL, CoVCE is about 12% higher than the acceptance ratio. Because the degree and closeness of topology and correlation property can effectively improve the acceptance ratio of virtual cluster requests.

Figure 10 shows the long-term revenue and cost overhead ratio. CoVCE has obtain a certain R/C advantage relative to the TM, IC and BL, because CoVCE measures the degree and closeness of topology information, and correlation property to evaluate the importance selection of the embedding and optimizes resource fragmentation, while CoVCE not only improves the virtual topology network acceptance ratio, but

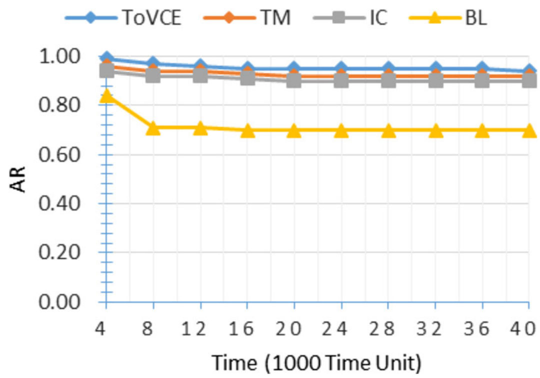

Fig. 8 VCR acceptance ratio over time comparison

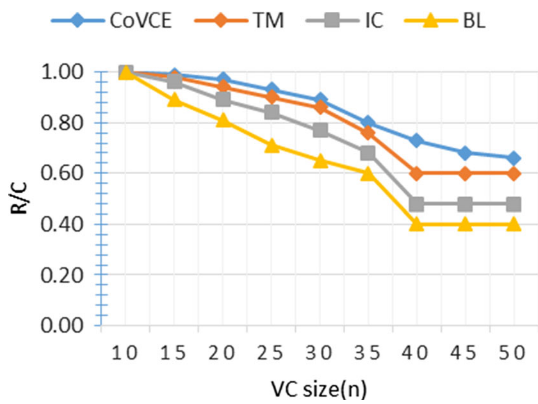

Fig. 9 Comparison of four algorithm R/C ratio

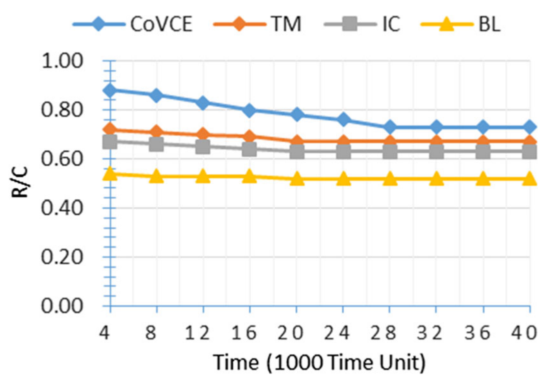

Fig. 10 Long-term R/C ratio comparison

also reduces the migration caused by improperly embedding virtual topology network and the remapping overhead. It is shown that CoVCE can obtain more revenues in Figs. 9 and 10, mainly because CoVCE improves the process of virtual node selection and reduces the remapping of virtual cluster requests due to the inappropriate node embedding of virtual cluster.

The network delay of controller-to-switch in virtual cluster is an important factor to affect the real-time performance of cluster system. By analyzing and comparing the influence of the algorithm on controller-to-switch delay, it is beneficial

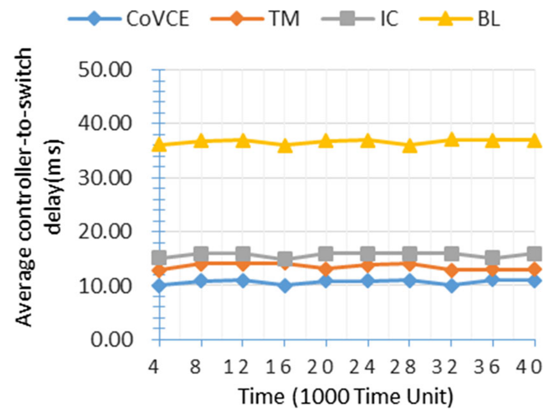

Fig. 11 Long-term average controller-to-switch delay

to the optimization of network delay. Taking into account the susceptible network delay of controller-to-switch, the virtual link embedding of controller-to-switch adopts the unsplittable k-shortest method to reduce the extra long-term average controller-to-switch delay, which is beneficial to approximate the actual product environment. The network latency, as shown in Fig. 11, is slightly better than TM and IC while it has a near 4 times network delay optimization compared to BL, because CoVCE takes into account the correlation property and the network centrality with the closeness and betweenness attributes.

7.2 Emulation

In order to approximate the real production environment, this paper emulates the average end-to-end network latency and throughput based on Mininet, FlowVisor and Opendaylight and analyzes the network latency and throughput by the network analysis tools ping, iperf (http://iperf.sourceforge.net/) and Wireshark (https://www.wireshark.org/#download). The nodes of virtual cluster are configured from 10 to 50 uniform distribution, and the other configuration is same to simulation parameters. Then, extensive experiments have been done to test the impact of the network delay and throughput with the scale size, respectively. Figure 12 shows the impact of the four algorithms on the average end-to-end network latency in virtual cluster. It can be seen that CoVCE (just choose k-shortest method for virtual link mapping in this emulation) has an advantage over other three algorithms in network latency, mainly because the virtual cluster is combined with the closeness and betweenness of the network centrality and the correlation property of network graph, resulting in a lower average network latency.

Figure 13 shows the impact of the four algorithms on the average throughput of the virtual cluster. CoVCE adopts the variance parameter to reduce the resource fragmentation, which leads to a certain unbalance of resource allocation

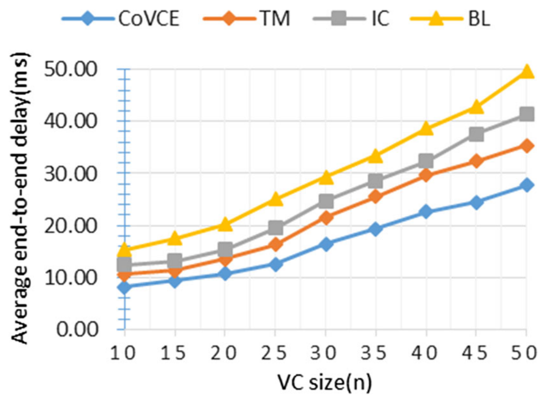

Fig. 12 Comparison of end-to-end delay over VC size

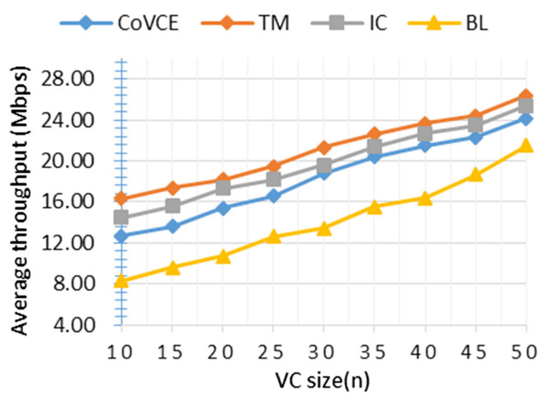

Fig. 13 Comparison of throughput over VC size

and results in a lower throughput than TM and IC. But in the real production environment, the pressure of nodes can be effectively balanced by adjusting the resource proportion parameter, such as limiting 80% resources to be allowed to allocate. Thus, the pressure of nodes is not too large, which can avoid virtual node performance degradation. In brief, CoVCE has achieved a better performance and obvious advantages in terms of acceptance rate, revenue, average controller-to-switch delay and throughout. Meanwhile, we explore the future research direction for the virtual cluster embedding in cloud data center.

8 Conclusion

This paper has explored a coordinated approach for virtual cluster embedding with VSDN controller based on the virtual cluster network topology through balancing network traffic flow and the efficiency of virtual cluster embedding. According to research on virtual network topology in cloud data center and combining the network centrality, the correlation property of the network graph and resource fragmentation, the heuristic algorithm, named CoVCE, has developed to adapt multiple factors of the coordinated virtual cluster embedding for optimizing the efficiency of VCE. Through theoretical and experimental analysis, this algorithm is verified to be able to optimize resource utilization effectively and to reduce network delays.

Acknowledgements This work is supported by National Core Electronic Devices, High-end Generic Chips and Basic Software Project under Grant No. 2016ZX01040101; Mobile Terminal Operating System under Grant No. 2016ZX01040101; the National Natural Science Foundation of China under Grant No. 61602492, 61502511, 61772070; Beijing Municipal Education Commission under Grant No.KM201510016009, Beijing Higher Education Young Elite Teacher Project (Grant No. YETP1169), Excellent Teachers Development Foundation of BUCEA (No.21082717046), and National Key R&D Program of China (No.2016YFC060090).

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

Informed consent Informed consent was obtained from all individual participants included in the study.

References

Al-Shabibi M, De Leenheer M, Gerola M, Koshibe A, Parulkar G, Salvadori E, Snow B (2014) OpenVirteX: make your virtual SDNs programmable. In: ACM SIGCOMM HotSDN workshop

Butt NF, Chowdhury M, Boutaba R (2010) Topology-awareness and reoptimization mechanism for virtual network embedding. Springer, Berlin

Cai J, Wang Y, Liu Y, Luo J-Z, Wei W, Xu X (2018) Enhancing network capacity by weakening community structure in scale-free network. Future Gener Comput Syst 87:765–771

Chen W, Chen Z, Samatova NF, Peng L, Wang J, Tang M (2014) Solving the maximum duo-preservation string mapping problem with linear programming. Theoret Comput Sci 530:1–11

Chowdhury M, Rahman M, Boutaba R (2012) Vineyard: virtual network embedding algorithms with coordinated node and link mapping. IEEE/ACM Trans Networking 20(1):206–219

Fischer A, Botero J, Beck M, De Meer H, Hesselbach X (2013) Virtual network embedding: a survey. IEEE Commun Surv Tutor 15:1–19

Gomes RL, Bittencourt LF, Madeira ER, Cerqueira E, Gerla M (2014) An architecture for dynamic resource adjustment in VSDNs based on traffic demand. In: GLOBECOM, pp 2005–2010

Gomes RL, Bittencourt LF, Madeira ER, Cerqueira E, Gerla M (2016) Bandwidth-aware allocation of resilient virtual software-defined networks. Comput Netw 100(5):179–194

Guan Z, Li J, Wu L, Zhang Y, Wu J, Du X (2017) Achieving efficient and secure data acquisition for cloud-supported internet of things in smart grid. IEEE Internet Things J 4(6):1934–1944

Guan Z, Si G, Zhang X, Wu L, Guizani N, Du X, Ma Y (2018) Privacy-preserving and efficient aggregation based on blockchain for power grid communications in smart communities. IEEE Commun Mag 56(7):1–7

He P, Deng Z, Wang H, Liu Z (2016) Model approach to grammatical evolution: theory and case study. Soft Comput 20(9):3537–3548

He P, Deng Z, Gao C, Wang X, Li J (2017) Model approach to grammatical evolution: deep-structured analyzing of model and representation. Soft Comput 21(18):5413–5423

Iperf-TCP and UDP bandwidth performance measurement tool. http://iperf.sourceforge.net/

Jhaveri R, Patel N, Zhong Y, Sangaiah A (2018) Sensitivity analysis of an attack-pattern discovery based trusted routing scheme for mobile ad-hoc networks in industrial IoT. IEEE Access. https://doi.org/10.1109/ACCESS.2018.2822945

Jiang L, Cheng Y, Yang L, Li J, Yan H, Wang X (2018) A trust-based collaborative filtering algorithm for E-commerce recommendation system. J Ambient Intell Humaniz Comput. https://doi.org/10.1007/s12652-018-0928-7

Lantz B, O'Connor B (2015) A Mininet-based virtual testbed for distributed SDN development. In: ACM conference on special interest group on data communication. ACM, pp 365–366

Li X, Wang H, Ding B, Li X, Feng D (2014a) Resource allocation with multi-factor node ranking in data center networks. Future Gener Comput Syst 32:1–12

Li J, Huang X, Li J, Chen X, Xiang Y (2014b) Securely outsourcing attribute-based encryption with checkability. IEEE Trans Parallel Distrib Syst 25(8):2201–2210

Li J, Chen X, Li M, Li J, Lee P, Lou W (2014c) Secure deduplication with efficient and reliable convergent key management. IEEE Trans Parallel Distrib Syst 25(6):1615–1625

Li R, Zhang J, Tan Y, Wu Q (2016) TMVCE—topology-aware multi-path virtual cluster embedding algorithm. In: CCIOT. IEEE

Li T, Chen W, Tang Y, Yan H (2018) A homomorphic network coding signature scheme for multiple sources and its application in IoT. Secur Commun Netw. https://doi.org/10.1155/2018/9641273

Lin W, Zhu C, Li J, Liu B, Lian H (2015) Novel algorithms and equivalence optimisation for resource allocation in cloud computing. IJWGS 11(2):193–210

Lin W, Xu S, Li J, Xu L, Peng Z (2017a) Design and theoretical analysis of virtual machine placement algorithm based on peak workload characteristics. Soft Comput 21(5):1301–1314

Lin W, Xu S, He L, Li J (2017b) Multi-resource scheduling and power simulation for cloud computing. Inf Sci 397:168–186

Medved J, Varga R, Tkacik A, Tkacik A, Gray K (2014) Opendaylight: towards a model-driven SDN controller architecture. In: 15th international symposium on 2014 IEEE. IEEE, pp 1–6

Mehmet D, Mostafa A (2014) Design and analysis of techniques for mapping virtual networks to software-defined network substrates. Comput Commun 45:1–10

Mijumbi R, Serrat J, Rubio J (2014) Dynamic resource management in SDN-based virtualized networks. In: Proceedings of international conference on network and service management. https://doi.org/10.1109/cnsm.2014.7014204

Pang Z, Wang K, Xie M (2014) The TH express-2 high performance interconnect networks. Front Comput Sci 8(3):357–366

Papagianni C, Androulidakis G, Papavassiliou S (2014) Virtual topology mapping in SDN-enabled clouds. NCCA

Rabbani M, Pereira Esteves R, Podlesny M, Simon G, Zambenedetti Granville L, Boutaba R (2013) On tackling virtual data center embedding problem. In: IFIP/IEEE IM 2013. Virtual network embedding base on real-time topological attributes, pp 177–184

Salvadori E, Doriguzzi Corin R, Broglio A, Gerola M (2011) Generalizing virtual network topologies in openflow-based networks. In: IEEE global telecommunications conference (GLOBECOM 2011), Houston, TX, USA, pp 1–6

Sun Z, Zhang Q, Li Y, Tan Y (2018) DPPDL: a dynamic partial-parallel data layout for green video surveillance storage. IEEE Trans Circuits Syst Video Technol 28(1):193–205

Tan Y, Xu X, Liang C, Zhang X, Zhang Q, Li Y (2018) An end-to-end covert channel via packet dropout for mobile networks. Int J Distrib Sens Netw 14(5):1550147718779568

Wang Z, Han Y, Lin T, Tang H, Ci S (2012) Virtual network embedding by exploiting topological information. In: IEEE GLOBECOM

Wang Y, Li K, Li K (2017) Partition scheduling on heterogeneous multicore processors for multi-dimensional loops applications. Int J Parallel Prog 45(4):827–852

Wang H, Wang W, Cui Z, Zhou X, Zhao J, Li Y (2018a) A new dynamic firefly algorithm for demand estimation of water resources. Inf Sci 438:95–106

Wang C, Shen J, Liu Q, Ren Y, Li T (2018b) A novel security scheme based on instant encrypted transmission for Internet-of-Things. Secur Commun Netw. https://doi.org/10.1155/2018/3680851

Wei X, Li H, Yang K, Zou L (2014) Topology-aware partial virtual cluster mapping algorithm on shared distributed infrastructures. IEEE Trans Parallel Distrib Syst 25(10):2721–2730

Wireshark. https://www.wireshark.org/#download

Yu M, Yi Y, Rexford J, Chiang M (2008) Rethinking virtual network embedding: substrate support for path splitting and migration. ACM SIGCOMM Comput Commun Rev 38(2):17–29

Yu X, Tan Y, Sun Z, Liu J, Liang C, Zhang Q (2018a) A fault-tolerant and energy-efficient continuous data protection system. J Ambient Intell Humaniz Comput. https://doi.org/10.1007/s12652-018-0726-2

Yu X, Zhang C, Xue Y, Zhu H, Li Y, Tan Y (2018b) An extra-parity energy saving data layout for video surveillance. Multimed Tools Appl 77:4563–4583

Yu X, Tan Y, Zhang C, Liang C, Khaled A, Zheng J, Zhang Q (2018c) A high-performance hierarchical snapshot scheme for hybrid storage systems. Chin J Electron 27(1):76–85

Zhang X, Tan Y, Xue Y, Zhang Q, Li Y, Zhang C, Zheng J (2017) Cryptographic key protection against FROST for mobile devices. Clust Comput 20(3):2393–2402

Zhang X, Tan Y, Zhang C, Xue Y, Li Y, Zheng J (2018) A code protection scheme by process memory relocation for android devices. Multimed Tools Appl 77(9):11137–11157

Zhou Z, Dong M, Ota K, Wang G, Yang LT (2016) Energy-efficient resource allocation for D2D communications underlaying cloud-RAN-based LTE-A networks. IEEE Internet Things J 3(3):428–438

Zhu H, Tan Y, Zhang X, Zhu L, Zhang C, Zheng J (2017) A round-optimal lattice-based blind signature scheme for cloud services. Future Gener Comput Syst 73:106–114

Zhu H, Tan Y, Yu X, Xue Y, Zhang Q, Zhu L, Li Y (2018) An identity-based proxy signature on NTRU lattice. Chin J Electron 27(2):297–303

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.