2019 IEEE International Conference on Services Computing (SCC)

# Enhancing Availability of Traffic-aware Virtual Cluster Allocation in Cloud Datacenters

Xuan Liu, Bo Cheng, Yi Yue, Meng Wang, Biyi Li, Junliang Chen
*Beijing University of Posts and Telecommunications*
*Beijing, China*
{*liuxuan0527, chengbo, yueyi, mengwang, lbyi0402, chjl*}*@bupt.edu.cn*

***Abstract*—As more and more services are deployed in the cloud datacenter, network traffic is growing exponentially. Virtual machines (VMs) of a virtual cluster (VC) must be allocated on physical machines (PMs) in the datacenter with a certain topology. Each VM need some resources to run various services. Apparently, allocating VMs in a VC as compactly as possible can reduce traffic consumption and avoid bandwidth-related bottlenecks. However, loose allocation scheme can reduce the loss expectation of VMs due to failure possibilities of PMs and switches, and the availability of the VC is increased thereby. To enhance availability and reduce network bandwidth usage, it is significant to determine the scheme of allocating VMs of the VC. In this paper, we first introduce four typical datacenter architectures with network topologies and corresponding cost matrices, and extend to generality. Then we propose a joint optimization function to measure risk of VC and core bandwidth usage with a global availability constraint. Subsequently, an evolution algorithm is raised to minimize the value of the constrained optimization function. Finally, the evaluation results show the effectiveness of the proposed approach and performance advancement over the existing approaches.**

***Keywords*-availability; traffic-aware; virtual cluster allocation; cloud datacenter**

## I. Introduction

Recent years have witnessed a rapid growth in the popularity of cloud computing, with an increasing trend towards more data-intensive applications in cloud datacenters. Many works have been done on designing datacenter network architectures, e.g. Fat-tree [1], VL2 [2] and BCube [3], etc. Those network architectures achieve various effects such as high availability, high flexibility, low communication cost. Generally, physical machines (PMs) are connected in the datacenter by switches, and all PMs and switches forms a specific network topology. In the datacenter, virtualization technology is used for higher hardware resource utilization of PMs. A data-intensive application is commonly deployed in a virtual cluster (VC). A VC contains a virtual switch, virtual links and multiple virtual machines (VMs) connected to the virtual switch by virtual links with bandwidth demands, and the VC always covers multiple services deployed over VMs on demand. Thus, network core traffic in the datacenter grows with more applications being deployed. The study in [2] shows that traffic caused by VMs can nearly take up 80% of the overall traffic in a cloud datacenter. In this case, allocating VMs as compactly as possible can save traffic consumption and avoid bandwidth-related bottlenecks.

However, compact allocation raises many concerns with respect to the availability, which is a significant issue to cloud applications and services. To be specifical, the PM failure results in the loss of all VMs allocated on it, while the switch failure leads to all PMs and VMs in the failure domain being unavailable. All communications between other VMs and those unavailable VMs are obstructed at the same time. One approach to mitigate the impact of the failures of PM/switch is to utilize redundant VMs and spread out all VMs in a VC for the application to as many different failure domains as possible. Thus, it is significant to find an optimal solution to VC allocation, to maximize the availability and minimize bandwidth usage in the datacenter.

Many fault-tolerant approaches have been proposed to enhance the availability of VC allocation [4]. Replication mechanism and checkpointing mechanism are two basic mechanisms. For replication mechanism, a direct way is to create more VM replicas of the VC and allocate all original VMs and VM replicas. Once a VM fails, the corresponding VM replica can replace the failed original VM or restore the original VM on an available domain. For checkpointing mechanism, the cardinal operation is to periodically save the execution state of the VM as a checkpointing image [5]. When a VM fails, the checkpointing image file can restore the VM from the last saved state and proceed services on the VM. These two mechanisms can handle failure events of the VM to enhance the availability of the VC. But more physical resources have to be taken in replica VMs and checkpointing image files. It is insufficient to just depend these mechanisms to provide the high availability for VC allocation and economize physical resources.

To solve these challenges, we propose an approach for enhancing the availability of the VC allocation (E-AVCA) with biogeography-based optimization (BBO) algorithm [6]. E-AVCA approach builds a risk model from the aspect of PM/switch failure and an available-VM rate model, aiming to enhance the availability of the VC allocation. Our approach minimizes the bandwidth consumption and the risk cost in the VC, and guarantees overall availability of VMs.

2474-2473/19/$31.00 ©2019 IEEE
DOI 10.1109/SCC.2019.00045

To summarize, this paper makes following contributions:

- We propose three novel models which are applicable to various cloud datacenters: the first one is to formulate the risk cost of violating the availability of the VC in terms of PM/switch failures; the second one is to quantify the availability-VM rate; the third one is to calculate the bandwidth usage of the VC by the communication cost matrix of the datacenter topology.
- Based on the above models, we formulate the availability-aware and traffic-aware VC allocation problem as a jointly constrained optimization problem. Then we introduce E-AVCA approach with BBO algorithm. The goal of E-AVCA approach is to minimize the function value to obtain the optimal solution for minimizing the overall risk cost and the bandwidth usage with all constraint conditions satisfied.

The rest of this paper is organized as follows. Section II presents the related work. Section III introduces system models and the problem statement in detail. Section IV develops the algorithm, followed by experimental results in Section V. Finally, Section VI concludes the paper.

## II. Related Work

The study [7] presents a high-level comprehensive survey on virtual machine placement (VMP) problem. We make some improvements based on the previous work, AVCA approach [8], which builds a risk model and a bandwidth usage model likewise. AVCA approach just considers the risk model and the bandwidth usage model on the fat-tree datacenter and focuses on the network traffic. Compared with AVCA approach, we can target VC allocation problems in different datacenters. We establish a global available-VM rate model to enhance the availability of VC allocation. And we attempt to effectively handle all constraints. These aspects are all lacking in the AVCA approach. The work in [9] also considers the risk cost against the availability of VMs, and minimizes the overall cost of risk and the energy consumption. But this work does not take the bandwidth usage into consideration. The work in [10] tackles traffic-aware VMP in datacenters with different topologies, and proposes a two-tier approximate algorithm based on balanced minimum k-cut (BMKC) method. The problem formulation assumes the number of VMs and the number of placement positions are equal, so that BMKC based on graph can conduct effectively, even for large-scale problem. At the same time, four typical datacenter network architectures are involved, namely tree, Fat-tree, VL2, and BCube. The evaluation shows that the proposed VMP algorithm can obtain greater benefit from a multi-tier architecture BCube, but less benefit from VL2 which adopts load balancing techniques. However, no QoS constraint is concerned in the work [10]. In our study, we also take these four types of architectures as instances to conduct our experiments. The work in the [11] addresses the reliability-aware VMP problem as placing at most $H$ groups of $k$ VMs on a minimum number of nodes. The proposed Integer Nonlinear Program method and a heuristic algorithm are combined to solve this problem. This approach can achieve good performance, but always needs larger running time than other heuristic approaches.

To heighten the reliability in cloud environment, A. Zhou [12] proposes a redundant VMP optimization approach that decides an appropriate placement of primary VMs and backup VMs with $k$-fault-tolerance assurance. The proposed heuristic algorithm aims to find a maximum weight matching based on bipartite graphs. The author also raises reliability by checkpointing mechanism in the cloud computing environment [13]. This work presents an optimal algorithm with edge switch failure-aware feature, in order to enhance the reliability of each failure domain in the datacenter.

In this paper, we extend AVCA problem and propose our E-AVCA approach which minimizes the risk value and the bandwidth usage. And E-AVCA approach is feasible for datacenters with different topologies and different scales.

## III. System Model and Problem Statement

In this section, we present models and the problem statement. To facilitate understanding following formulations, TABLE I clarifies all notations throughout this section.

Table I: Notations

| Symbol | Meaning |
|---|---|
| $M$ | the number of switches |
| $N$ | the number of PMs |
| $P_s$ | failure probability of the switch $s$ |
| $p_k$ | failure probability of the PM $k$ |
| $C_{N*N}$ | communication cost matrix of PMs |
| $c_{ij}$ | communication cost from PM $i$ to PM $j$ |
| $P(E_1\|s)$ | probability of Event 1 of switch $s$ |
| $P(E_2\|s)$ | probability of Event 2 of switch $s$ |
| $\delta_I$ | probability proportionate of Event 1 |
| $\delta_{II}$ | probability proportionate of Event 2 |
| $n$ | the number of VMs in the VC |
| $B_{n*n}$ | bandwidth demand matrix of VMs |
| $b_{ij}$ | bandwidth demand between VM $i$ and VM $j$ |
| $X$ | VC allocation scheme and $X = \{x_1, x_2, ..., x_n\}$ |
| $x_i$ | target PM of VM $i$ 's allocation and $x_i \in \{1, 2..., N\}$ |
| $f(x_i, k)$ | function for judging the matching of VM $i$ and PM $k$ |
| $g(x_i, s)$ | function for judging the matching of VM $i$ and switch $s$ |
| $\varphi_{I,s}$ | the number of failed VMs when Event1 of switch $s$ occurs |
| $\varphi_{II,s}$ | the number of failed VMs when Event2 of switch $s$ occurs |
| $\Phi_s$ | the number of available VMs when switch $s$ fails |
| $\mu$ | expectation of the number of available VMs |
| $\sigma$ | standard deviation of the number of available VMs |
| $Risk$ | risk cost of violating the availability with a VC allocation |
| $\eta$ | available-VM rate |
| $\eta_0$ | lower limit of $\eta$ |
| $BW$ | whole bandwidth usage |
| $Risk_0$ | the minimum risk value |
| $Risk_\infty$ | the maximum risk value of the VC allocation |
| $BW_0$ | the minimum bandwidth usage |
| $BW_\infty$ | the maximum bandwidth usage of the VC allocation |
| $\theta$ | weight parameter of the joint optimization function |
| $D_i^x$ | CPU/memory/disk demand of VM $i$, $x \in$ {CPU,Mem,Disk} |
| $C_k^x$ | CPU/memory/disk capacity of PM $k$, $x \in$ {CPU,Mem,Disk} |

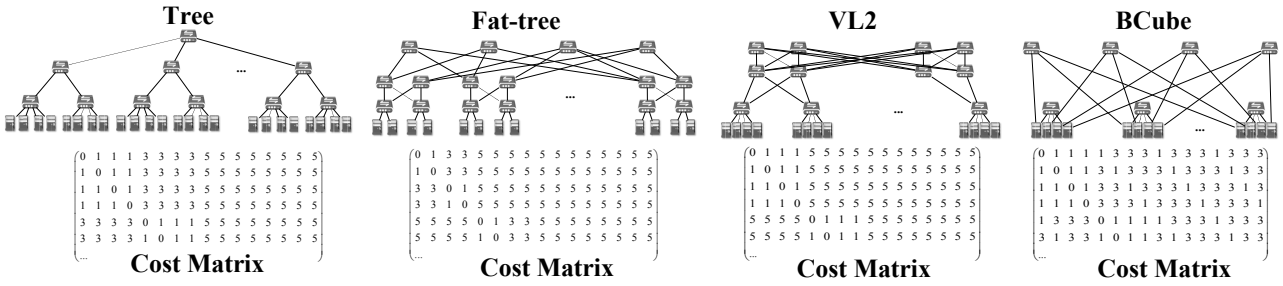


Figure 1: Network topologies and corresponding cost matrices for four datacenter network architectures

### A. Datacenter Network Architectures

In Fig.1, four types of network architectures are presented with corresponding cost matrices for traffic load, as the study [10] defines. Then we briefly introduce datacenters with tree, fat-tree, VL2, and BCube topologies, respectively.

*1) Tree:* Tree is a common datacenter topology, with a vertical connection of multiple levels of star structures.

*2) Fat-tree:* Fat-tree is a three-tier architecture and is built around the notion of pods: a collection of edges and aggregation switches that composes a complete bipartite graph. In this way, Fat-tree architecture realizes the timely processing of the load through multiple links in the core layer to avoid network hotspots, and to avoid overload issues by feasibly dividing the traffic within the pod.

*3) VL2:* VL2 is a new architecture proposed by Microsoft in 2009 [2]. This kind of architecture enables datacenters to save the cost while ensuring flexible and dynamic allocation of resources. And VL2 supports uniform high capacity between servers in the datacenter.

*4) BCube:* BCube is a new multi-level network architecture which is specifically designed for shipping container-based modular datacenters. Those servers sharing the same position in different recursive units in the same layer are connected to the same upper layer switch.

We assumes that a datacenter has $M$ switches and $N$ PMs. The matrix $C_{N*N}$ shows the connectivity of all PMs in the datacenter. And $c_{ij}$ denotes the number of switches that are required for data transmission from PM $i$ to PM $j$.

### B. Virtual Cluster Model

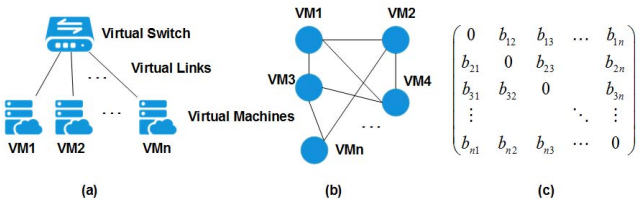


Figure 2: (a)VC abstraction. (b)VMs topology in a VC. (c)Bandwidth demand matrix of VMs.

As Fig. 2-(a) shows, we exploit the hose model to build the VC abstraction. In this abstraction, a VC consists of a virtual switch, $n$ VMs and respective virtual links with the virtual switch. The VC model is designed for all-to-all communication pattern and is fit for processing big data in data-intensive applications. A data-intensive application may cover multiple services, which are deployed over different VMs on demand. Therefore, each VM has its own bandwidth demand with other VMs. From this perspective, as shown in Fig, 2-(b) and (c), all VMs in the VC can form a connected graph with a bandwidth demand matrix $B_{n*n}$. And $b_{ij}$ denotes the bandwidth demand between VM $i$ and VM $j$.

### C. Availability Model

We consider the situations of PM/switch failure as:

- Event 1: there is no PM fails when switch $s$ fails.
- Event 2: there is a PM fails when switch $s$ fails.
- Event 3: there is a PM fails but there is no switch fails.

Since a switch failure will result in all VMs allocated in corresponding PMs unavailable, we build the risk model via Event 1 and Event 2. When no switch fails, a single PM failure will directly make all VMs on it unavailable. So we quantify the available-VM rate $\eta$ in the case of Event 3 subsequently. $Risk$ and $\eta$ are combinatorially taken into consideration as the basis of E-AVCA problem.

*1) Risk:* Suppose the failure probability of switch/PM is independent. The equation group 1 shows probabilities of occurrence for Event 1 and Event 2, where $p_i$ denotes the failure rate of PM $i$, and $N$ represents the number of PMs. Denote $P(E|s)$ as the sum of $P(E_1|s)$ and $P(E_2|s)$.

$$\begin{cases} P(E_1|s) = \prod_{k=1}^{N}(1-p_k) \\ P(E_2|s) = \sum_{k=1}^{N} p_k \prod_{\substack{l=1 \\ l\neq k}}^{N}(1-p_l) = \sum_{k=1}^{N} \frac{p_k}{1-p_k} P(E_1|s) \\ P(E|s) = P(E_1|s) + P(E_2|s) \end{cases} \tag{1}$$

For a given VC allocation scheme, the vector $X = \{x_1, x_2, ..., x_n\}$ presents the allocation of each VM in the VC. More specifically, the relation $x_i = k$ represents that VM $i$ is allocated on the PM $k$. The function $f(x_i, k)$ is given for judging the matching of VM $i$ and PM $k$. Similarly,

the function $g(x_i, s)$ is given for judging the matching of VM $i$ and switch $s$. The equation $f(x_i, k) = 1$ holds if and only if the equation $x_i = k$ holds. And the equation $g(x_i, s) = 1$ if and only if host $x_i$ is connected to switch $s$.

We denote $\delta_I$ as the probability proportionate of Event 1, as presented in the equation 2. When Event 1 occurs, the number of unavailable VMs in the VC $\varphi_{I,s}$ can be computed as the equation 3 shows.

$$\delta_I = \frac{P(E_1|s)}{P(E|s)} \tag{2}$$

$$\varphi_{I,s} = \delta_I \sum_{i=1}^{n} g(x_i, s) \tag{3}$$

Similarly, we denote $\delta_{II}$ as the probability proportionate of Event 2, as presented in the equation 4. To obtain the number of unavailable VMs in the VC $\varphi_{II,s}$, we should consider two cases. In one case, the failed PM is linked to the switch $s$. In the other case, the failed PM is not. So $\varphi_{II,s}$ can be computed as the sum of these two parts, as shown in the equation 5.

$$\delta_{II} = \frac{P(E_2|s)}{P(E|s)} \tag{4}$$

$$\varphi_{II,s} = \delta_{II}\Big(\sum_{i=1}^{n} g(x_i, s) + \sum_{k=1}^{N}(1 - g(k, s))p_k \sum_{i=1}^{n} f(x_i, k)\Big) \tag{5}$$

When switch $s$ fails, the available VM $\Phi_s$ can be obtained by the equation 6, where $n$ is the number of VMs in the VC. With the given VC allocation scheme, the expectation of the number of available VMs $\mu$ can be obtained in the equation 7, where $P_s$ is the failure probability of switch $s$. The standard deviation $\sigma$ can be obtained in the equation 8.

$$\Phi_s = n - (\varphi_{I,s} + \varphi_{II,s}) \tag{6}$$

$$\mu = n - \sum_{s=1}^{M} P_s(\varphi_{I,s} + \varphi_{II,s}) \tag{7}$$

$$\sigma = \sqrt{\frac{1}{M-1}\sum_{s=1}^{M}(\Phi_s - \mu)^2} \tag{8}$$

Therefore, the range of practical number of available VMs in the VC is $[\mu - \sigma, \mu + \sigma]$. A large ratio of $\sigma$ to $\mu$ obviously will fluctuate with the availability of the VC allocation. As the equation 9 shows, we use the $Risk$ parameter to define the risk cost of violating the availability with a specific VC allocation scheme.

$$Risk = \frac{\sigma}{\mu} = \frac{\sqrt{\frac{1}{M-1}\sum_{s=1}^{M}\left[(1 - \sum_{s=1}^{M} P_s)(\varphi_{I,s} + \varphi_{II,s})\right]^2}}{n - \sum_{s=1}^{M} P_s(\varphi_{I,s} + \varphi_{II,s})} \tag{9}$$

*2) Available-VM rate $\eta$:* In the case of Event 3, that is, when a single PM $k$ fails, the ratio of available VMs in all VMs in the VC $\eta_k$ is computed as shown in equation 10. Based on the study [14], the reliability of a VC can be defined as the available ratio of VMs with a worst-case of a server failure. Likewise, we define the $\eta$ parameter as the minimum value among all $\eta_k$. The available-VM rate $\eta$ is directly quantified and applicable as a constraint for the VC allocation, to enhance the availability in terms of Event 3.

$$\eta : \min\left\{\eta_k = \frac{n - \sum_{i=1}^{n} f(x_i, k)}{n} = 1 - \frac{1}{n}\sum_{i=1}^{n} f(x_i, k)\right\} \tag{10}$$

### D. Bandwidth Consumption Model

The whole bandwidth usage model with a specific VC allocation can be built by matrices $B_{n*n}$ and $C_{N*N}$, which are the bandwidth demand matrix of VMs in the VC and the communication cost matrix of PMs in the datacenter, respectively. Thus, the bandwidth usage can be formulated as the equation 11 shows, where $n$ denotes the number of VMs in the VC, $x_i$ and $x_j$ denote the PMs which accommodate VM $i$ and VM $j$, respectively.

$$BW = \sum_{i=1}^{n-1}\sum_{j=i}^{n}(b_{ij} \cdot c_{x_i x_j}) \tag{11}$$

### E. Jointly Constrained Optimization Function

We combine those two objectives, the equation 9 and the 11, into a joint optimization function with the normalization pattern. As shown in the equation 12, $Risk_0$ and $Risk_\infty$ are the minimum value and the maximum value of $Risk$, respectively; $BW_0$ and $BW_\infty$ are the minimum value and the maximum value of bandwidth usage, respectively; and $\theta$ is a tunable parameter in $(0, 1)$. These constraints in the equation 13 include necessary conditions of CPU, memory and disk resources. That means, the sum of resource requirements of all VMs in the VC must be less than the candidate-allocating PM's capacity. Moreover, $\eta$ must be no less than a given $\eta_0$.

$$\min F(X) = \theta\frac{Risk - Risk_0}{Risk_\infty - Risk_0} + (1 - \theta)\frac{BW - BW_0}{BW_\infty - BW_0} \tag{12}$$

$$\text{s.t.} \begin{cases} \sum_{i=1}^{n} f(x_i, k) \cdot D_i^{CPU} < Cap_k^{CPU} \\ \sum_{i=1}^{n} f(x_i, k) \cdot D_i^{Mem} < Cap_k^{Mem} \\ \sum_{i=1}^{n} f(x_i, k) \cdot D_i^{Disk} < Cap_k^{Disk} \\ \eta \geq \eta_0 \end{cases} \tag{13}$$

Consequently, we consider E-AVCA problem as a constrained optimization problem based on our models.

## IV. ALGORITHM

Biogeography-based optimization (BBO) algorithm [6] is a natural heuristic algorithm for solving optimization problems. The idea of BBO algorithm is derived from the distribution and migration of biological populations in habitats. Biological populations live in different habitats and have different Habitat Suitability Indexes (HSI) in different habitats. HSI is related to rainfall, vegetation diversity, landform characteristics, climate, and so on. These characteristic variables form a Suitability Index Variables (SIVs) vector that describes the habitat suitability. The suitability of a habitat is inversely proportional to populations.

In this section, we briefly introduce BBO algorithm at first. Then we elaborate the combination of E-AVCA problem with BBO algorithm. Finally we propose the implementation scheme of E-AVCA approach with BBO algorithm to solve the joint optimization problem with constraints. To facilitate understanding the algorithm better, TABLE II shows all mathematical notations throughout this section.

Table II: Notations

| Symbol | Meaning |
|---|---|
| $s$ | the number of populations, position number of a habitat |
| $n$ | the maximum number of populations, the number of processed habitats in each iteration |
| $E, I$ | coefficients of migration functions |
| $\lambda_s$, $\mu_s$ | immigration/emigration rate of population $s$ |
| $P_s$ | the probability that the habitat holds $s$ populations |
| $P_{max}$ | the maximum probability in all $P_s$ |
| $m_s$ | mutation probability of population $s$ |
| $m_{max}$ | the maximum mutation probability |
| $h_i(X,k)$ | $i$-th constraint expression |
| $\alpha_i$ | penalty multiplier of $h_i$ |
| $G$ | the number of generations |
| $E_n$ | the number of elite solution in each iteration |
| $D$ | dimension of the SIVs vector |
| $H$ | candidate habitats set |

### A. BBO Algorithm

In the BBO algorithm, different habitats represent different candidate solutions to the optimization problem. HSI is related to the optimization function value, and represents the quality of each candidate solution. A higher HSI in the optimization problem indicates a better solution, and a lower HSI indicates a worse solution. A SIVs vector represents a candidate solution. Migration operation and mutation are two core operations on SIVs vectors: migration operation refers to the probabilistic sharing of information between habitats; mutation operation refers to habitat changes due to emergencies. With migration operation and mutation operation, different habitats can obtain different suitability values. Finally, BBO algorithm will achieve dynamic balance and obtain the optimal solution to the optimization problem.

*1) Migration Operation:* The migration operation on a single habitat is described as a model: the immigration rate $\lambda_s$ and the emigration rate $\mu_s$ are functions of population $s$ on the habitat. The maximum number of populations that the habitat can accommodate is $n$. Fig.3 presents four types of migration models with different $\lambda_s$ and $\mu_s$. When $s$ increases, $\lambda_s$ and $\mu_s$ gradually decreases and increases, respectively. There is an equilibrium $s_0$ in the condition of $\lambda_s = \mu_s$. For each habitat, a new habitat can be generated after the migration operation for each SIV. More SIVs can be inherited from the habitat with a relatively high HSI. So more SIVs vectors with close distance to the habitat with higher HSI can be searched. Then new habitats will be ready for the following mutation operation. Mutation probability of each habitat is related to population $s$, and migration probability $\lambda_s$ and $\mu_s$.

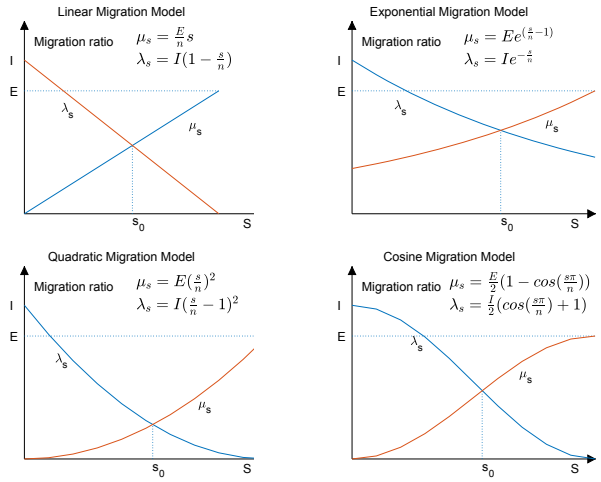


Figure 3: Four Migration Models of BBO

Assume $P_s$ as the probability that the habitat accommodates exactly $s$ populations. Equation 14 shows the function model of $P_s$ from $t$ to $t+\Delta t$. When $\Delta t \to 0$, the equilibrium value of $P_s$ is computed as shown in the equation 15.

$$\begin{aligned} P_s(t+\Delta t) = (1-\lambda_s\Delta t - \mu_s\Delta t)P_s(t) \\ + \lambda_{s-1}\Delta t P_{s-1} + \mu_{s+1}\Delta t P_{s+1} \end{aligned} \tag{14}$$

$$P_s = \begin{cases} \dfrac{1}{1+\sum\limits_{k=1}^{n}\frac{\lambda_0\lambda_1\cdots\lambda_{k-1}}{\mu_1\mu_2\cdots\mu_k}} & s=0 \\ \dfrac{\lambda_0\lambda_1\cdots\lambda_{s-1}}{\mu_1\mu_2\cdots\mu_s\left(1+\sum\limits_{k=1}^{n}\frac{\lambda_0\lambda_1\cdots\lambda_{k-1}}{\mu_1\mu_2\cdots\mu_k}\right)} & 1\le s\le n \end{cases} \tag{15}$$

*2) Mutation Operation:* As natural disasters and diseases can completely change the ecological environment of a habitat, all populations of the habitat will be out of equilibrium. BBO algorithm uses a mutation operation to simulate this phenomenon. The mutation probability $m_s$ can be formulated as shown in the equation 16, where $P_{max}$ represents the max value of $P_s$, and the constant $m_{max}$ represents the upper limit of the mutation probability.

$$m_s = m_{max}(1 - P_s/P_{max}) \tag{16}$$

### B. Combination of E-AVCA Problem and BBO Algorithm

For the E-AVCA problem, we make some correspondences and improvements with BBO algorithm.

We transform all constraints in the equation 13 as constraint functions as follows. Then we formulate $F^*(X)$ as shown in the equation 18, with penalty multipliers $\alpha_i >> 1$. In this way, minimizing $F^*(X)$ is the optimization objective without any constraints via penalty functions.

$$\begin{cases} h_1(X,k) & = \sum\limits_{i=1}^{n} f(x_i,k) \cdot D_i^{CPU} - Cap_k^{CPU} \\ h_2(X,k) & = \sum\limits_{i=1}^{n} f(x_i,k) \cdot D_i^{Mem} - Cap_k^{Mem} \\ h_3(X,k) & = \sum\limits_{i=1}^{n} f(x_i,k) \cdot D_i^{Disk} - Cap_k^{Disk} \\ h_4(X,k) & = \eta_0 - \eta \end{cases} \tag{17}$$

$$F^*(X) = F(X) + \sum\nolimits_i \alpha_i \cdot \sum\nolimits_k [\max(h_i(X,k),0)]^2 \tag{18}$$

The number of populations is $n$. For each population $s$, $\lambda_s$, and $\mu_s$ can be obtained with a specific migration model (linear migration model is adopted in our approach). SIVs vectors are candidate solutions to the VC allocation problem. The mutation operation can randomly modify SIVs based on $P_s$ and $m_{max}$. The value of $m_{max}$ is tunable.

### C. Implementation Scheme of E-AVCA Approach

As shown in the Algorithm 1, we present the implementation scheme of E-AVCA approach with BBO algorithm to solve the joint optimization problem. Lines 1 to 5 are initialization, then the optimization loop begins. In addition, $BW_0$ can be approximately acquired by the IVCA algorithm [15]. And three assistant algorithms in the work [8], VCMBW, VCMiR, and VCMaR, can approximately compute $BW_\infty$, $Risk_0$, and $Risk_\infty$, respectively. Lines 8 to 17 are migration operation, using $\lambda_s$ and $\mu_s$ to decide how much information to share between habitats probabilistically. Lines 18 to 24 are mutation operation, with $\mu_s$ to decide which SIVs need to be updated. Finally, habitats should be sorted by HSI. When the iterative process ends, the final solution is obtained.

**Algorithm 1** E-AVCA approach with BBO algorithm

**Input:** PMList, switchList, communication cost matrix $C$, VMList, bandwidth demand matrix $B$, $\eta_0$
**Output:** VC allocation solution $X$

```
1: Initialize parameters from PMList, switchList, VMList
2: Initialize other parameters of BBO algorithm
3: Compute λ_s, μ_s, and m_s
4: Compute Risk_0, Risk_∞, BW_0, and BW_∞, respectively
5: Initialize and sort a random set of habitats H by HSI in
   ascend, which is computed by F*(X)
6: for g = 1 to G do
7:    Save E_n best habitats
8:    for i = 1 to s_n do
9:       if rand(0,1) < λ_i then
10:         for j = 1 to D do
11:            if rand(0,1) < μ_j then
12:               Select a SIV randomly from the habitat H_j
13:               Replace the SIV in H_i with the selected SIV
14:            end if
15:         end for
16:      end if
17:   end for
18:   for i = 1 to s_n do
19:      for j = 1 to D do
20:         if rand(0,1) < m_j then
21:            Randomly replace the j-th SIV of H_i
22:         end if
23:      end for
24:   end for
25:   Sort all habitats by HSI
26:   Replace E_n habitats at the end with the elites
27:   Sort all habitats by HSI again
28: end for
29: return VC allocation solution X from H
```

## V. Experimental Evaluations

### A. Experiment Settings

We conduct our experiments in CloudSim [16] environment, and datacenters with tree, Fat-tree, VL2 and BCube architectures are all involved. There are 1024 PMs in the datacenter, and 64 or 128 edge switches with different topologies. Each PM is randomly set to have a dual-core CPU with performance equivalent to 3720 or 5320 MIPS, 4GB of RAM and 1.0 TB of disk storage [17]. The failure rates of switches and PMs are set randomly in $[0.05, 0.2]$ and $[0.02, 0.12]$ [8]. Each VC includes 128 VMs with a certain topology. The CPU, memory and disk storage capacity of each VM is chosen randomly from: 500 MIPS, 0.6 GB, 1.0 GB; 1000 MIPS, 1.7 GB, 1.0 GB; 2000 MIPS, 3.75 GB, 1.0 GB; 2500 MIPS, 0.85 GB, 1.0 GB [17]. The range of the available-VM rate $\eta$ is $[95.3\%, 99.2\%]$, which can be computed by the equation 10. We set the lower limit of available-VM rate $\eta_0$ at $96.8\%$. The bandwidth demand of each interconnected VM pair is randomly set in [100,500] Mbps. Those parameters in the Algorithm 1 are set as follows: the population size is set 50; the number of generation is set at 500; the number of elite solutions is set at 2; the maximum mutation probability is set at 0.1 [18].

### B. Effectiveness of E-AVCA Approach Analysis

Fig. 4 shows those minimum optimization function values in datacenters with tree, Fat-tree, VL2, and BCube topologies, respectively. With the iteration of generation, these optimal results in different datacenters all tend to be stable. From the perspective of the minimum function value, all final results are in $(0.20, 0.50)$. That means, E-

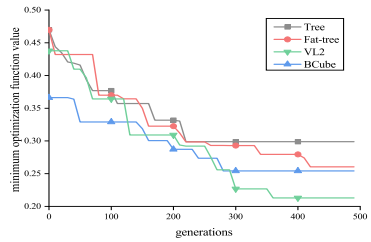


Figure 4: Minimum cost value

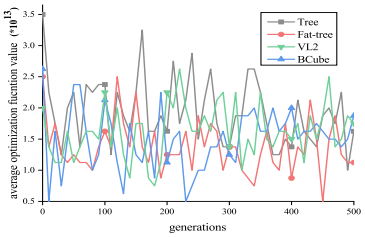


Figure 5: Average cost value

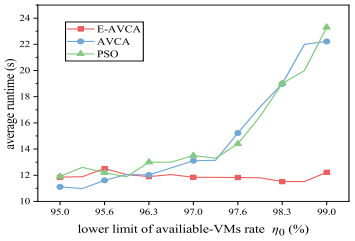


Figure 6: Average runtime

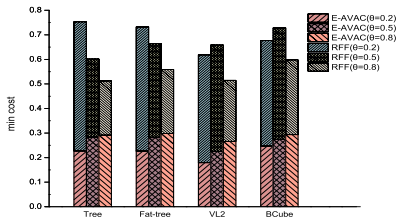


Figure 7: Joint function values

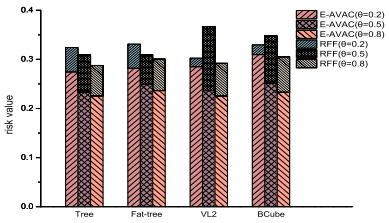


Figure 8: Risk cost values

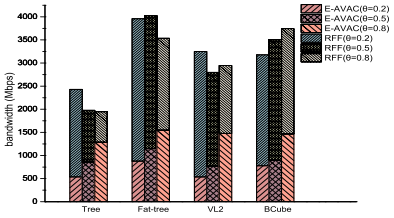


Figure 9: Bandwidth usages

AVCA approach can obtain effective VC allocation solutions that can simultaneously optimize the availability and the bandwidth usage of the VC. And those average optimization function values of different datacenters are shown in Fig. 5. Since all constraint functions are added to the optimization function by penalty multipliers, the problem becomes an unconstrained optimization problem. The order of magnitude is $10^{13}$ in the average function value, and is far greater than that of the minimum function value. Those solutions that violate any constraint condition will be discarded in the iteration process. Results shown in Fig. 4 and Fig. 5 can reveal the effectiveness of E-AVCA approach.

### C. Comparison of Average Runtime

We compare E-AVCA approach with AVCA approach [8] and PSO approach [19], which are both population-based optimization approaches. We conduct comparison experiments in the Fat-tree datacenter, which is the precondition of modeling in the AVCA approach. Since AVCA approach does not consider the available-VM constraint, we follow the way that AVCA approach handles other resource constraints to complement the processing of this constraint. We set $\eta_0$ in $[95.0\%, 99.0\%]$. As shown in Fig. 6, when $\eta_0 \geq 97.6\%$, AVCA approach and PSO approach will spend more runtime with the increase of $\eta_0$. These two approaches both take inspection of constraints after obtaining VC allocation solutions, and re-enter the iteration process or check other elite solutions if the solution violates any constraint. The advantage of E-AVCA approach is that E-AVCA approach considers all constraints in the process of iteration. The runtime result of E-AVCA approach is stable with different $\eta_0$ constraints, so E-AVCA approach can be more efficient.

### D. Analysis of Optimization Result

We analyse the impact of the tunable parameter $\theta$ in different datacenters and $\theta$ is set at 0.2, 0.5 and 0.8. Meanwhile, we compare E-AVCA approach with Random First-fit (RFF) approach [8] to evaluate the optimization results. RFF approach preferentially meets all constraints and then obtain an effective VC allocation scheme. So RFF approach is suitable for VC allocation problems in the datacenters with pervasive topologies. As Fig. 7 to Fig.9 shows, RFF can obtain a VC allocation solution with all constraints satisfied, but with little optimization of the availability and the bandwidth usage. In this case, RFF approach is a classical greedy approximation approach and can be a reference value used to analyse the optimization rate of E-AVCA approach. We use the method to compute the optimization rate as follows:

$$\textit{Optimization Rate} = \frac{\textit{Value(RFF)} - \textit{Value(E-AVCA)}}{\textit{Value(RFF)}} \quad (19)$$

When $\theta = 0.2, 0.5, 0.8$, average optimization rates of the joint optimization function value are $68.34\%$, $59.67\%$, and $47.18\%$ respectively. With the increase of weight the factor $\theta$, the risk value decreases and the bandwidth usage increases. For these two objectives, corresponding optimization results are: average optimization rates of the risk value are $10.50\%$, $26.89\%$, and $22.38\%$, respectively; average optimization rates of the bandwidth usage are $78.60\%$, $68.72\%$, and $50.17\%$, respectively.

In terms of the datacenter architecture, average optimization rates in the tree, Fat-tree, VL2 and BCube topologies are $55.25\%$, $57.78\%$, $61.67\%$, and $58.89\%$, respectively. VL2 datacenter can obtain more optimization due to its topological features. Specifically, the bandwidth usage difference $BW_\infty - BW_0$ in the VL2 topology is larger than the difference in other topologies. We can find that in the VL2 communication cost matrix, most of PM pairs will take higher cost value. Thus, there will be a bigger optimization range in terms of the bandwidth usage of VL2 datacenters.

## VI. Conclusion

In this paper, we proposed an approach for enhancing the availability of traffic-aware VC allocation in cloud datacenters. We formally described an availability model and a bandwidth consumption model. In the availability model, situations of PM/switch failure were both considered. In this case, a risk model and an available-VM rate model were built. Based on these models, we formulated E-AVCA problem as a jointly constrained optimization problem. We proposed the E-AVCA approach with BBO algorithm, and leveraged both theoretical demonstration and experimental evaluations on different settings. Finally, evaluation results showed the effectiveness and efficiency of our approach.

For the future work, we will integrate some mechanisms to enhance the reliability of VMs, such as VM backups and VM migration. With constrained conditions, and optimization objectives involved, the improved VC allocation problem will take more challenges and be more meaningful.

## Acknowledgment

This work was supported in part by the National Natural Science Foundation of China (No.61772479), and the National Key Research and Development Program of China (2017YFB1400603).

## References

[1] R. Niranjan Mysore, A. Pamboris, N. Farrington, N. Huang, P. Miri, S. Radhakrishnan, V. Subramanya, and A. Vahdat, "Portland: A scalable fault-tolerant layer 2 data center network fabric," in *Proceedings of the ACM SIGCOMM 2009 Conference on Data Communication*, ser. SIGCOMM '09, 2009, pp. 39–50.

[2] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta, "Vl2: A scalable and flexible data center network," in *Proceedings of the ACM SIGCOMM 2009 Conference on Data Communication*, ser. SIGCOMM '09, 2009, pp. 51–62.

[3] C. Guo, G. Lu, D. Li, H. Wu, X. Zhang, Y. Shi, C. Tian, Y. Zhang, and S. Lu, "Bcube: A high performance, server-centric network architecture for modular data centers," in *Proceedings of the ACM SIGCOMM 2009 Conference on Data Communication*, ser. SIGCOMM '09, 2009, pp. 63–74.

[4] M. Treaster, "A survey of fault-tolerance and fault-recovery techniques in parallel systems," *CoRR*, vol. abs/cs/0501002, 2005.

[5] A. Zhou, Q. Sun, and J. Li, "Enhancing reliability via checkpointing in cloud computing systems," *China Communications*, vol. 14, no. 7, pp. 1–10, 2017.

[6] D. Simon, "Biogeography-based optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 6, pp. 702–713, 2008.

[7] Z. Á. Mann, "Allocation of virtual machines in cloud data centersa survey of problem models and optimization algorithms," *Acm Computing Surveys (CSUR)*, vol. 48, no. 1, p. 11, 2015.

[8] J. Liu, S. Wang, A. Zhou, F. Yang, and R. Buyya, "Availability-aware virtual cluster allocation in bandwidth-constrained datacenters," *IEEE Transactions on Services Computing*, pp. 1–1, 2018.

[9] Z. Yang, L. Liu, C. Qiao, S. Das, R. Ramesh, and A. Y. Du, "Availability-aware energy-efficient virtual machine placement," in *2015 IEEE International Conference on Communications (ICC)*, June 2015, pp. 5853–5858.

[10] X. Meng, V. Pappas, and L. Zhang, "Improving the scalability of data center networks with traffic-aware virtual machine placement," in *2010 Proceedings IEEE INFOCOM*. IEEE, 2010, pp. 1–9.

[11] S. Yang, P. Wieder, R. Yahyapour, S. Trajanovski, and X. Fu, "Reliable virtual machine placement and routing in clouds," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 10, pp. 2965–2978, Oct 2017.

[12] A. Zhou, S. Wang, B. Cheng, Z. Zheng, F. Yang, R. N. Chang, M. R. Lyu, and R. Buyya, "Cloud service reliability enhancement via virtual machine placement optimization," *IEEE Transactions on Services Computing*, vol. 10, no. 6, pp. 902–913, Nov 2017.

[13] A. Zhou, Q. Sun, and J. Li, "Enhancing reliability via checkpointing in cloud computing systems," *China Communications*, vol. 14, no. 7, pp. 1–10, July 2017.

[14] Xuemin Wen, Yanni Han, Bing Yu, X. Chen, and Zhen Xu, "Improving revenue for reliability-aware vdc embedding in a software-defined data center," in *2016 IEEE/ACM 24th International Symposium on Quality of Service (IWQoS)*, June 2016, pp. 1–2.

[15] J. Liu, S. Wang, A. Zhou, S. A. P. Kumar, F. Yang, and R. Buyya, "Using proactive fault-tolerance approach to enhance cloud service reliability," *IEEE Transactions on Cloud Computing*, vol. 6, no. 4, pp. 1191–1202, Oct 2018.

[16] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. D. Rose, and R. Buyya, "Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Softw., Pract. Exper.*, vol. 41, no. 1, pp. 23–50, 2011.

[17] A. Beloglazov and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers," *Concurrency and Computation: Practice and Experience*, vol. 24, no. 13, pp. 1397–1420, 2012.

[18] H. Ma, S. Ni, and M. Sun, "Equilibrium species counts and migration model tradeoffs for biogeography-based optimization," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, Dec 2009, pp. 3306–3310.

[19] S. Wang, Z. Liu, Z. Zheng, Q. Sun, and F. Yang, "Particle swarm optimization for energy-aware virtual machine placement optimization in virtualized data centers," in *2013 International Conference on Parallel and Distributed Systems*, Dec 2013, pp. 102–109.