2014 IEEE 6th International Conference on Cloud Computing Technology and Science

Virtual Network Allocation for Fault Tolerance with Bandwidth Efficiency in a Multi-Tenant Data Center

Yukio Ogawa*, Go Hasegawa† and Masayuki Murata‡
*Telecommunications & Network Systems Division, Hitachi, Ltd., Tokyo, Japan, Email: yukio.ogawa.xq@hitachi.com
†Cybermedia Center, Osaka University, Osaka, Japan, Email: hasegawa@cmc.osaka-u.ac.jp
‡Graduate School of Information Science and Technology, Osaka University, Osaka, Japan, Email: murata@ist.osaka-u.ac.jp

***Abstract*—In a multi-tenant data center, nodes and links of tenants' virtual networks (VNs) share a single component of the physical substrate network (SN). A failure of the single SN component can thereby cause simultaneous failures of multiple nodes and links in a VN; this complex of failures must significantly disrupt the services offered on the VN. In the present paper, we clarify how the fault tolerance of a VN is affected by a SN failure, especially from the perspective of VN allocation in the SN. We propose a VN allocation model for multi-tenant data centers and formulate a problem that deals with the bandwidth loss in the VN due the SN failure. We conduct numerical simulations with the setting that has 1.7×10^8 bit/s bandwidth demand on each VN. The results show that the bandwidth loss can be reduced to 5.3×10^2 bit/s per VN, but the required bandwidth between physical servers in the SN increases to 1.0×10^9 bit/s per VN when each node in the VN is mapped to an individual physical server. The balance between the bandwidth loss and the required bandwidth between physical servers can be optimized by assigning every four nodes of the VN to each physical server, meaning that we minimize the bandwidth loss without providing too sufficient bandwidth in the core area of the SN.**

***Keywords*-data center; multi-tenant; virtual network allocation; multiple simultaneous failures; fault tolerance**

I. Introduction

A data center for the Infrastructure-as-a-Service (IaaS) type of cloud computing serves virtual data center infrastructures for client organizations, i.e., tenants. In order to host not only business-critical applications but also mission-critical ones in a virtual infrastructure, high availability must be ensured through the use of a fault-tolerant design. One of the typical methods for building the virtual infrastructure is to introduce an overlay network architecture based on a tunneling protocol such as VXLAN (Virtual Extensible Local Area Network) [1]. In this architecture, the virtual network (VN) for a tenant is built as an overlay network by connecting VN nodes, i.e., virtual machines (VMs), that are pooled on the physical servers of the physical substrate network (SN) at the data center. Although the topology of the VN is independent on that of the SN, the components of the VN should be appropriately assigned to those in the SN in order to share the SN's resources effectively and tolerate SN failures. This paper focuses on clarifying the fault tolerance of the VN in terms of the influence from SN failures and establishing an efficient allocation of resources to the VN. Our goal is thereby to ensure high availability for the VN so that mission critical applications can be hosted on it.

Mapping VNs to the shared SN in the data center faces similar issues when embedding them in a shared ISPs (Internet Service Providers) network. These issues are commonly referred to as the *Virtual Network Embedding* (VNE) problem, which has been a major research topic in network virtualization [2]. In the VNE problem, the availability, survivability, and resiliency of VNs have been improved by minimizing the network disconnections and capacity loss due to physical link failures [3], [4] and by minimizing the sum of all working and backup resources of physical nodes and links [5], [6]. Similar to VNEs, a number of proposals have been made on reliability-aware resource allocation and redundancy provisioning in data center networks. One is an allocation scheme that aims at minimizing the impact of failures on VNs by spreading out the VMs across multiple fault-domains while reducing bandwidth consumption in the core area of the SN [7]. Another is a scheme that considers minimum shared backup resources reserved on physical links and nodes after physical failures [8]–[10]. The latter studies suppose that the VN allocation has an impact on the SN resource consumption in terms of backup and restore resources after SN failures. These studies, however, do not consider fault tolerance of the VN. Although a fault-tolerance metric has been proposed in [7], failure-recovery characteristics and bandwidth loss during the recovery time are out of its scope.

In this paper, we focus on the performance characteristic of concurrent VNs sharing resources of the SN. In detail, we clarify how physical resource allocation to VN nodes and links affects the fault tolerance of the VN. For this purpose, we first hypothesize that the recovery time of the VN increases with the failure complexity and explain our procedure for controlling this recovery time. Second, we propose a model of a multi-tenant data center network and formulate a problem that deals with the impact of a failure in the SN, expressed in terms of bandwidth loss in the VN. Third, by applying a heuristic method to solve

978-1-4799-4093-6/14 $31.00 © 2014 IEEE
DOI 10.1109/CloudCom.2014.64

the problem, we examine how much the VN allocation affects the bandwidth loss on failure. We describe that the bandwidth loss in a VN is highly dependent on whether components in the VN are consolidated in a few physical components or distributed to many physical components. We also show the trade-off between the bandwidth loss and the bandwidth required by the VN in the SN.

The rest of this paper is organized as follows. In Section II, we present our hypothesis about the VN recovery time. In Section III, we describe a model of a multi-tenant network and the problem expressing our allocation scheme. Section IV presented a heuristic for solving the problem. In Section V, we evaluate the VN allocation, and finally, in Section VI, give conclusions and discuss remaining issues.

II. A Hypothesis on the Failure Recovery Time

In a virtualized environment, many VNs are consolidated into the SN for better physical resource utilization as well as cost effectiveness. Multiple components of VNs thereby share, e.g., a single physical server in the SN. As a result, a single failure of the server can simultaneously disrupt multiple nodes and links in a VN. This characteristic significantly impacts the availability of the VN, as compared to a traditional network composed of dedicated physical components. Previous studies have shown that multiple simultaneous failures in a network can lead to a longer recovery time [11] as a result of, e.g., the complexity of fault localization in large enterprise systems [12], and SRG (Shared Risk Group) failures in optical networks [4]. On the basis of our knowledge and experience, we believe that the same problems exist in VNs in a multi-tenant data center.

Based on the above discussion, the hypothesis for the relationships between the failure complexity and the network recovery time is illustrated in Figure 1. When the complexity of failures in a VN is low, e.g., when only a few components in the VN fail simultaneously, the VN can recover after a few seconds by automatically switching to hot-standby nodes and links. This can be done by using existing autonomous decentralized control techniques such as VRRP (Virtual Router Redundancy Protocol) [13], and VMware FT (Fault Tolerance) [14]. We call this technique *hot-standby recovery*. On the other hand, if the complexity of the failure is high, the VN has a great risk of inducing unexpected behaviors from misconfigurations, software bugs, etc. [15], [16]. This behavior can prevent the failed nodes and links from switching to standby ones. It can thereby significantly delay the restoration of the VN, which may end up taking several minutes or even hours. Since this type of failures depends on the implementation and configuration of the VN, it is difficult to predict the occurrence of such failures as well as their duration in advance. We hence adopt a centralized control technique by which the failed nodes are forced to be terminated and cold-standby nodes are alternatively booted [17], [18]; this technique can reduce the network downtime to a few minutes. We call this technique *cold-standby recovery*. As explained above, we propose a procedure for reducing this downtime; this procedure combines hot- and cold-standby recovery and switches from the former to the latter with reference to the failure complexity.

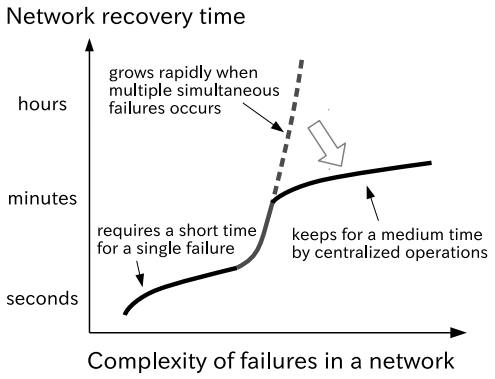


Figure 1. Hypothesis of time to recovery from failures in a single VN

III. Virtual Network Allocation

Here, we propose a multi-tenant network model and a VN allocation scheme for tolerating SN failures.

A. Network Model in a Multi-tenant Environment

Figure 2 gives an overview of mapping a VN onto an underlay SN. In what follows, we give models of the VN and the SN, both of which are defined including end nodes, i.e., end VMs and physical servers, respectively.

The SN is modeled with the following sets and parameters.

$G_u = (V, W, E)$: a SN topology consisting of a set of end nodes (i.e., physical servers) V, a set of intermediate nodes (i.e., physical switches) W, and a set of physical links E. The identifiers of a physical server, a physical switch, and a physical link are v, w, and e, respectively.

P : a set of physical paths between pairs of physical servers. The identifier of a physical path is p.

R_v $(v \in V)$: a constant number of CPU cores of physical server v.

S_e $(e \in E)$: a constant bandwidth of physical link e.

D_v, D_w, D_e $(v \in V, w \in W, e \in E)$: constant failure rates (the number of failures per device per unit time) of physical server v, physical switch w, and physical link e, respectively.

Furthermore, a physical path p is mapped onto a physical server v, physical switch w, and physical link e by using the following parameters.

$\gamma_{pv}, \gamma_{pw}, \gamma_{pe} \in \{0, 1\}$ $(p \in P, v \in V, w \in W, e \in E)$: binary variables that take a value of 1 if the physical path p is mapped onto physical server v, physical

switch w, and physical link e, respectively, and 0 otherwise. The values of γ_{pv} and γ_{pw} depend on the value of γ_{pe}.

Note that though each physical server accesses external storage via a storage area network (SAN), we will omit the SAN because it hardly affects the VN allocation in our model.

Now, let I be the set of VNs hosted on the SN. The ith VN is defined as follows.

$G_o^i = (N_i, L_i)$: the ith VN topology consisting of a set of logical nodes (i.e., VMs) N_i and a set of logical links L_i. The identifiers of a VM and a logical link are n and l, respectively.

F_i : a set of traffic flows between pairs of end VMs. The identifier of a traffic flow is f.

r_n^i $(n \in N_i)$: a constant number of CPU cores required by VM n in the ith VN.

c_f^i $(f \in F_i)$: a constant average bandwidth of traffic flow f in the ith VN. Moreover, the constant total average bandwidth for accessing the services offered on the ith VN from an external network is defined as C_i $(C_i = \sum_{f \in F_i} c_f^i)$.

Furthermore, in the ith VN, traffic flow f is assigned to logical link l by using the following parameters.

$\delta_{fl}^i \in \{0,1\}$ $(i \in I, f \in F_i, l \in L_i)$: binary variables that take the value of 1 if traffic flow f is routed through logical link l and 0 otherwise.

In the multi-tenant model, node n and link l of the ith VN are mapped onto physical server v and path p of the SN, respectively. They are defined by the following binary variables.

$x_{lp}^i \in \{0,1\}$ $(i \in I, l \in L_i, p \in P)$: binary variables that take the value of 1 if link l of the ith VN is mapped onto physical path p and 0 otherwise.

$x_{nv}^i \in \{0,1\}$ $(i \in I, n \in N_i, v \in V)$: binary variables that take the value of 1 if node n of the ith VN is mapped onto physical server v and 0 otherwise.

We also introduce the following attributes related to the VN allocation.

T_v^i, T_w^i, T_e^i $(i \in I, v \in V, w \in W, e \in E)$: the recovery time periods of the ith VN after a failure happens on physical server v, physical switch w, and physical link e, respectively. These are explained in Section III-C.

B. Problem Description

In this paper, the goal of VN allocation is to minimize the bandwidth loss when a failure happens in the SN. Let B_i denote the bandwidth loss of the ith VN.

Objective: minimize

$$\sum_{i \in I} B_i = \sum_{i \in I} \left(\sum_{v \in V} B_v^i + \sum_{w \in W} B_w^i + \sum_{e \in E} B_e^i \right), \qquad (1)$$

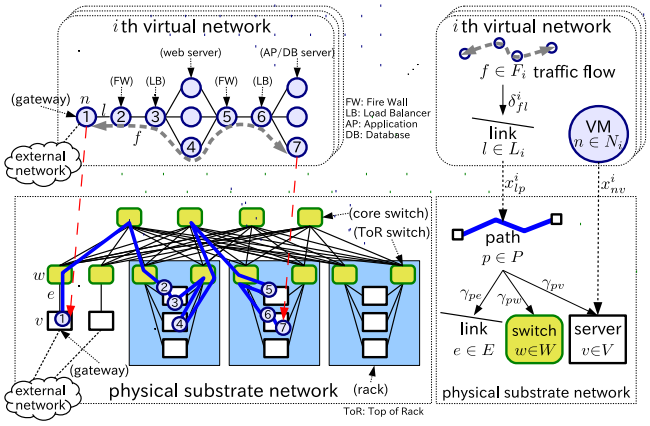


Figure 2. Network model for a multi-tenant data center

where B_v^i, B_w^i, and B_e^i are the bandwidth losses of the ith VN resulting from a failure of physical server v, physical switch w, and physical link e, respectively. These variables are defined as

$$B_v^i = D_v T_v^i \sum_{f \in F_i} X_{fv}^i c_f^i, \qquad (2)$$

$$B_w^i = D_w T_w^i \sum_{f \in F_i} X_{fw}^i c_f^i, \qquad (3)$$

$$B_e^i = D_e T_e^i \sum_{f \in F_i} X_{fe}^i c_f^i, \qquad (4)$$

where X_{fv}^i, X_{fw}^i and X_{fe}^i are binary variables that respectively indicate the assignments of the ith VN's traffic flow f to physical server v, physical switch w, and physical link e. Each of these variables is given below by using the notation $\bigcup$, which means logical sum here.

$$X_{fv}^i = \bigcup_{p \in P} \bigcup_{l \in L_i} \gamma_{pv} x_{lp}^i \delta_{fl}^i, \qquad (5)$$

$$X_{fw}^i = \bigcup_{p \in P} \bigcup_{l \in L_i} \gamma_{pw} x_{lp}^i \delta_{fl}^i, \qquad (6)$$

$$X_{fe}^i = \bigcup_{p \in P} \bigcup_{l \in L_i} \gamma_{pe} x_{lp}^i \delta_{fl}^i, \qquad (7)$$

Objective (1) states that the bandwidth loss of each VN is the sum of the losses caused by the failures of physical servers, physical switches and physical links. Equation (2) means that the bandwidth loss in the VN resulting from physical server failures is the product of the VN's failure rate and total amount of lost traffic, both of which are caused by a physical server failure. B_w^i in Equation (3) and B_e^i in Equation (4) are calculated in the same manner. Note that we do not consider multiple simultaneous failures in the SN,

because the probability of multiple failures is much smaller than that of a single failure in the SN. We also assume that a failure in the VN does not spread to other VNs.

The constraints on Objective (1) are stated below.

Subject to:

$$\sum_{p \in P} x_{lp}^{i} = 1, \quad \forall l \in L_i, i \in I \tag{8}$$

$$\sum_{i \in I} \sum_{n \in N_i} x_{nv}^{i} r_n^{i} \leq a_1 R_v, \quad \forall v \in V \tag{9}$$

$$\sum_{i \in I} \sum_{p \in P} \gamma_{pe} \sum_{l \in L_i} x_{lp}^{i} \sum_{f \in F_i} \delta_{fl}^{i} c_f^{i} \leq a_2 S_e, \quad \forall e \in E \tag{10}$$

Constraint (8) ensures that a logical link is certainly assigned to a physical path; this assignment implies both end nodes of the link are embedded, too. Constraint (9) ensures that the ratio of the sum of the CPU cores required by the VMs assigned to a physical server to the CPU cores on the physical server is less than a_1. Constraint (10) ensures that the ratio of the bandwidth sum of the traffic flows going through a physical link to the bandwidth provided by the physical link is less than a_2. Here, a_1 satisfies $0 < a_1 < 1$, and it guarantees each physical server provides CPU cores for standby VMs after any single failure in the SN. a_2 also satisfies $0 < a_2 < 1$, and it ensures that each physical link carries fail-over traffic after a failure in the SN.

C. Modeling Recovery Time of a Virtual Network

On the basis of the hypothesis in Section II, we present a model for recovery time periods of the ith VN, i.e., T_v^i, T_w^i, and T_e^i, after failures of the physical server v, physical switch w, and physical link e in Equations (2), (3) and (4). As explained in Section III-A, a VM and a link in the VN are mapped onto a physical server and a physical path in the SN, respectively. In this type of data center network, when a failure occurs in a physical switch/link and disrupts the VN links mapped onto the switch/link, recovery of the physical switch/link leads to recovery of the VN links. The VN itself does not have a capability of recovering from such failures and relies on the recovery mechanism (e.g., equal-cost multi-path routing) of the SN [19]. On the other hand, if a failure occurs in a physical server and it has an impact on a VN having VMs embedded in the server, the VN should recover the VMs by utilizing its own failure-recovery mechanism.

In accordance with the above mechanisms, T_w^i and T_e^i are approximately equal to the recovery time of the physical switch w and physical link e, respectively. T_w^i and T_e^i are thus defined as constants not influenced by how the ith VN is embedded in physical switch w and link e. By contrast, T_v^i depends on how complicated the ith VN becomes as a result of the failure of physical server v. We can express the level of complexity as the number of multiple VMs failing simultaneously in the ith VN as a result of a failure of physical server v; this number is equivalent to the number of VMs in the ith VN that have been assigned to the physical server v. Let m_v^i denote this number. As explained in Section II, the ith VN recovers from the failure by performing the procedure below.

- Each of the VMs in the ith VN is paired with a dedicated hot-standby VM in advance. When a VM fails, the paired hot-standby VM takes over in the case of $m_v^i \leq \theta$, where θ is a threshold parameter.
- Redundant shared resources are set aside for cold-standby VMs in each physical server. When a VM in the ith VN fails, an alternative cold-standby VM is booted to succeed it in the case of $m_v^i > \theta$.

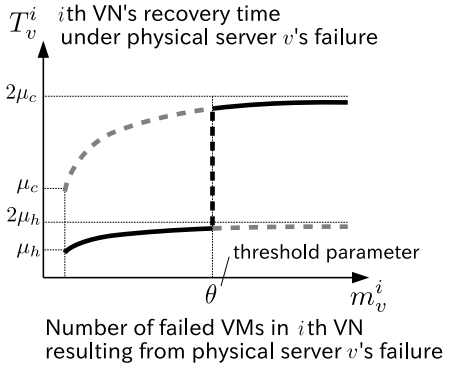


Figure 3. Recovery time model of a single VN

T_v^i is thus modeled as follows. In general, the service time distribution commonly used in computer systems is an exponential distribution (with mean μ and variance μ^2) [20]. We assume that each VM's processing time to recover from a physical server failure also follows this distribution. T_v^i is defined as the maximum recovery time among m_v^i VMs; this recovery time is approximated as the sum of the mean and the standard deviation time among m_v^i VMs. We assume that m_v^i VMs fail coincidentally and recover independently of each other. The expected standard deviation of the m_v^i VMs' recovery delays is thus $\sqrt{\frac{m_v^i - 1}{m_v^i}}\,\mu$. We define that the m_v^i VMs recover after a mean time μ_h through hot-standby recovery in the case of $m_v^i \leq \theta$, or after a mean time μ_c through cold-standby recovery otherwise. T_v^i is consequently defined as (see Figure 3).

$$T_v^i = \begin{cases} \left(1 + \sqrt{\frac{m_v^i - 1}{m_v^i}}\right) \mu_h & (m_v^i \leq \theta) \\ \left(1 + \sqrt{\frac{m_v^i - 1}{m_v^i}}\right) \mu_c & (m_v^i > \theta) \end{cases} \tag{11}$$

As explained above, if more than θ VMs in the VN are assigned to a physical server, these VMs will recover through cold-standby recovery. Because the VN has to prioritize its fault tolerance and needs to shorten the recovery time of the VMs, the following constraint is added to those

in Section III-B and used in the evaluations in Section V.

Subject to:

$$\sum_{n \in N_i} x_{nv}^i \leq \theta, \quad \forall i \in I, \forall v \in V \tag{12}$$

Constraint (12) prohibits the VN from assigning more than θ VMs to a physical server, thereby ensuring that the VMs will recover only through hot-standby recovery.

IV. Algorithm for Virtual Network Allocation

The VN allocation problem explained in Section III-B is a sort of VN embedding problem [2], which reduces to an multi-way separator problem that is *NP*-hard to solve optimally [21]. We thus propose a heuristic approach to solve it. The heuristic is composed of a two-stage allocation procedure. Upon receiving a tenant system request, a VN is initially allocated according to a *Greedy Algorithm* [22]. Then, the initial allocation is refined by a *Tabu search* [23].

The first stage utilizes the fact that a physical server connected to an external network (called a gateway server) allocates its resources only to a VM connected to the external network (called a gateway VM) (see Figure 2). We thereby begin by assigning a gateway VM to a corresponding gateway server. A link connecting an assigned VM and an unassigned VM is then iteratively mapped so as to minimize Objective (1), until all of the VMs and the links in the VN are assigned. The second stage repeatedly moves each of the assigned VMs in the VN to another physical server to find a better assignment in the neighborhood of the current placement, until all possible assignments are checked or the number of VM replacements is above a threshold. These two algorithms try to assign a shorter physical path to a logical link in the VN in order to reduce the bandwidth consumed in the core of the SN.

V. Evaluation

Here, we describe the trade-off between the fault-tolerance and the SN resource usage.

A. Multi-Tenant Data Center Network for Evaluation

A VN was configured to have an active-active topology for a mission-critical application; half of this VN topology is shown in the upper left of Figure 2. Each node in half of the VN was paired with a dedicated node in the other half for the purpose of hot-standby recovery. Corresponding to the VN topology, the SN components (i.e., gateway servers, core switches, Top-of-Rack (ToR) switches, and physical servers in each rack shown in lower left of Figure 2) were divided into two symmetrical parts. Both halves of the SN allocated their resources to corresponding halves of the VN. As both allocations were exactly the same, we will only explain half of the allocations.

The SN had a two-level fat-tree topology configured as an non-oversubscribed network (see Figure 2). The SN used a 32-port switch for both the ToR and core switches; this SN thereby consisted of 8 core switches, 16 ToR switches, and 120 physical servers in its maximum configuration. Each rack included a ToR switch and 8 physical servers. The number of CPU cores in each physical server, R_v, was 32, and the bandwidth of each link, S_e, was 1.0×10^{10} bit/s. a_1 in Constraint (9) and a_2 in Constraint (10) were set to 0.9 and 0.5, respectively; these values were determined by considering fail-over after any single physical failure. Under the above settings, 3,360 CPU cores were made available for allocation.

Table I
Failure rate (in failures per year) of SN components

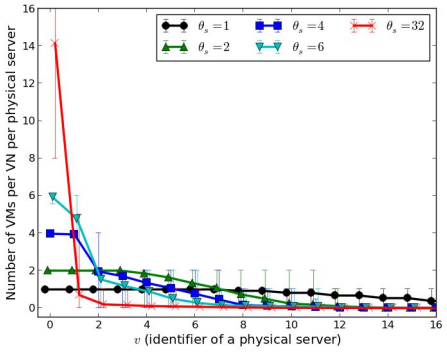
D_v	2.53 [25], 3, 6 [26]
D_w	0.005 - 0.111 [15], 0.055 - 0.073 [24]
D_e	0.054, 0.095 [15], 0.073 [26]

The VN was modeled after the traditional three-tier web serving architecture illustrated in the upper left of Figure 2. To make the VN model simple, the numbers of web servers and application (AP)/database (DB) servers in the VN were set to the same value; each number followed a truncated normal distribution with mean 5, standard deviation 3 and lower limit 2. Under these settings, each VN had 5.8 web and AP/DB servers and a total of 15.7 VMs (except for the gateway VM) on average. The traffic flow was defined per path routed through a pair of web and AP/DB servers and had an average bandwidth of 3.0×10^7 bit/s. Average bandwidth for accessing the services offered by these servers in the VN from an external network, C_i, was thereby 1.7×10^8 bit/s. Moreover, the number of CPU cores required by each VM, r_n^i, was set to 1. The hot-standby recovery time of a VM, μ_h, was set to 4 s, and the cold-standby recovery time, μ_c, was set to 60 s.

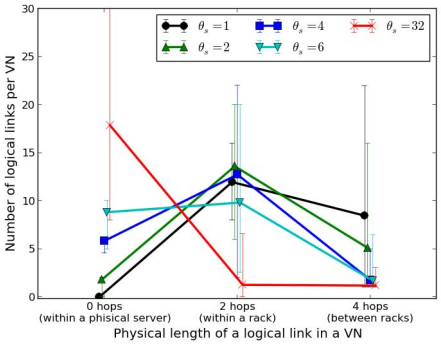
There have been several studies on network failures [15], [24]–[26]. Table I summarizes the failure rate (in failures per device per year) of a single physical server, D_v, that of a single physical switch, D_w, and that of a single physical link, D_e, and their source references. Note that the higher failure rate of D_v is due to software-related errors of the operating systems and hypervisors [25], [26]. In our evaluations, D_v was set to 4, and D_w and D_e were neglected because they are much smaller than D_v. Objective (1) thus depended on the first term alone.

B. Evaluation Results

To evaluate Objective (1), the fault-recovery time is given by Equation (11), in which the threshold θ must take various values from 1 to 32 depending on many factors such as processes, configurations, software implementations of VMs in the VN. Here, the maximum value, 32, was set to be equal to R_v/r_n^i. In actual operations, it is difficult to define the value of θ in advance. We therefore initially chose a value



(a) Assignment of VMs



(b) Assignment of links

Figure 4. VN assignment

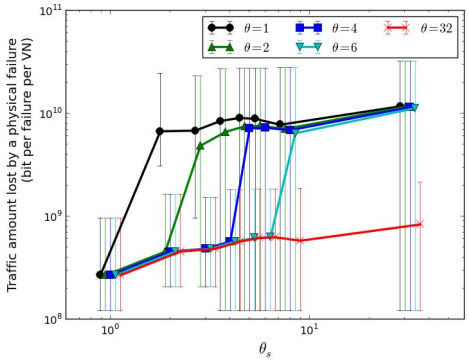
of θ, θ_s, and allocated VNs so as to minimize Objective (1) by using θ_s. Then, we evaluated the allocation for various values of θ. Note that horizontal positions of the plots in this section have been adjusted to keep the error bars visible.

1) Overview of a VN Mapping: Figure 4 shows the overview for the VN assignment, where the identifiers on the horizontal axis are sorted in descending order and each marker specifies the mean and each error bar specifies the 5% and 95% values for all VN allocations. When θ_s was set to 32, almost all of the VMs in the VN, except for the gateway VM, were consolidated in a single physical server. Most of the links in the VN were virtually assigned within the physical server and did not occupy the bandwidth of the physical links. In contrast, when θ_s was set to 1, each VM was mapped onto an individual physical server, and each link was mapped onto a physical link between two physical servers. As θ_s became smaller, the VMs became distributed to more physical servers due to Constraint (12), and thereby, more bandwidth of the physical links became occupied. The recovery time T_v^i was assumed to increase drastically due to simultaneous failures of more than θ_s VMs. This resulted in mapping θ_s or less VMs in the VN onto a single physical server. As explained above, the value of θ_s determines the *shape* of the VN; i.e., it determines whether the VN is one with VMs and logical links scattered across many physical servers and physical links, or one consolidating all VMs and links in a few physical servers.

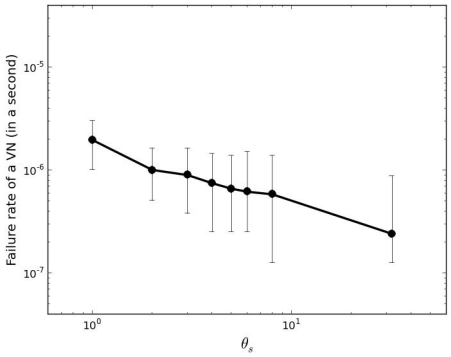
2) Trade-Off between Fault Tolerance and Physical Bandwidth Consumption: In order to evaluate the relationship between the shape and fault tolerance of the VN, we analyzed how the optimized Objective (1) changed with θ_s, as well as θ. Figure 5(a) shows the average traffic amount in the VN lost by a physical failure (bit per failure per VN), which corresponds to $T_v^i \sum_{f \in F_i} X_{fv}^i c_f^i$ in Eq. (2). If θ_s was set to 1, the traffic amount was the smallest (2.7×10^8 bit) for any θ. Because the VMs were each distributed to an individual physical server, the traffic flows spread across many physical servers so as to minimize the traffic amount lost by one failure of a physical server. As θ_s increased, the traffic amount increased, as a result of consolidating more VMs and thereby flowing more traffic into a single physical server. Although the traffic amount lost by a failure was also smaller (less than 10^9 bit) for $\theta \geq \theta_s$ at that time, it increased significantly (around 10^{10} bit) for $\theta < \theta_s$ (resulting from cold-standby recovery). When θ_s was set to the maximum value of 32, the traffic amount reached the maximum for any θ, except for $\theta = \theta_s$. Figure 5(b) shows the average failure rate of the VN, which corresponds to $\sum_{v \in V} D_v \sum_{f \in F_i} X_{fv}^i$ related to Eq. (2). This failure rate decreased from 2.0×10^{-6} to 2.4×10^{-7}, when θ_s was set to a large value and more VMs and traffic flows in a VN were concentrated in fewer physical servers.

The average bandwidth lost by a physical failure in the VN, B_i in Objective (1), (Figure 5(c)) was affected by both the traffic amount lost by a physical failure and the failure rate. When θ_s was set to 1 and each VM was distributed to an individual physical server, the average B_i was nearly the minimum (5.3×10^2 bit/s per VN) for any θ. This is because each VM used hot-standby recovery even though the failure rate of the VN was high. When θ_s became large and VMs were consolidated in fewer physical servers, the average B_i slightly decreased, as long as $\theta \geq \theta_s$. This is because the decrease in the VN failure rate had more influence than the increase in the traffic amount lost by a physical failure. However, if θ was smaller than θ_s, the average B_i significantly increased to around 6.0×10^3 bit/s per VN because each VM used cold-standby recovery. When θ_s was set to a large value like 32, all the VMs in the VN were concentrated in a single physical server and the failure rate of the VN decreased. In this case, cold-standby recovery was applied unless $\theta \geq \theta_s$. As a result, the average B_i reached the maximum value for any θ other than $\theta \geq \theta_s$

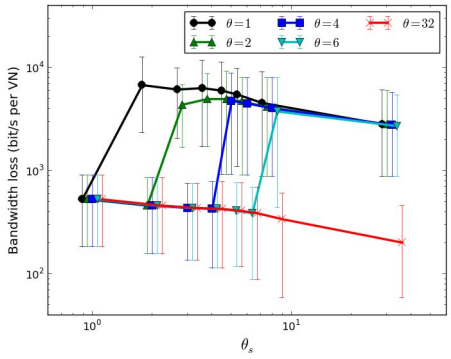
In order to describe the influence of the VN's shape on the requirements for the SN, we analyzed how θ_s changed the bandwidth used by each VN outside the physical servers (Figure 6). When θ_s was set to 1 and most of the logical links in the VN were mapped to physical links between



(a) Traffic amount lost by a physical failure



(b) Failure rate



(c) Loss of bandwidth

Figure 5. Fault tolerance of each VN

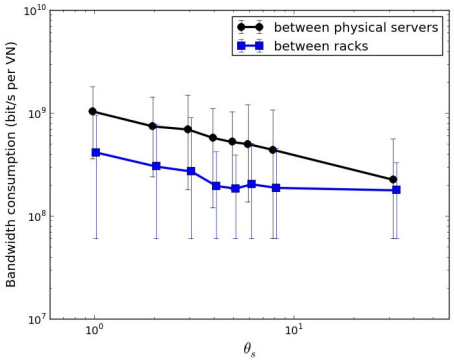


Figure 6. Physical bandwidth consumed by each VN

and within racks, both the average consumed bandwidth between servers and that between racks reached the maximum (between servers: 1.0×10^9 bit/s per VN, between racks: 4.2×10^8 bit/s per VN). As θ_s increased and more logical links were consolidated in a physical server, both bandwidths became smaller. The smallest bandwidth between racks (about 2.0×10^8 bit/s per VN) was when θ_s was 4; here, the VN had almost no inter-rack traffic flows other than the one coming through the gateway. In this case, almost all of the logical links were mapped onto the physical links between the physical servers and ToR switches. The traffic flows from a VM went to and back from the ToR switch in a rack and were not forwarded outside the rack. Moreover, when θ_s was set to the maximum, 32, all of the logical links except for the one connected to the gateway were embedded in a few physical servers; this minimized both bandwidths.

3) VN Allocation Policy Derived from the Results: As shown in Figures 5(c) and 6, the risk of bandwidth loss in each VN caused by a physical failure increases with θ_s and the bandwidth consumed by the VN's usual traffic flows decreases with θ_s. We should consider which θ_s is applicable to actual operations. Although we must reduce the risk of significant service disruptions caused by multiple simultaneous failures in the VN, the excess capacity required for fault tolerance should be kept as low as possible. One of the best approaches is therefore to minimize the bandwidth loss of the VN resulting from sharing physical resources while avoiding holding too many redundant core switches. This state is called Pareto optimality [27]. Under our evaluation settings, this was achieved when θ_s was 4.

VI. Conclusion

We described the fault tolerance of each VN in an IaaS type of data center, focusing on the situation of multiple simultaneous failures in each VN caused by a single physical failure. Through numerical evaluations based on our hypothesis about the VN recovery time, we found the following results. We set the average bandwidth of 1.7×10^8 bit/s for accessing the services offered on each VN in advance. When each of the VMs in the VN was mapped to an individual physical server, the bandwidth loss fell to 5.3×10^2 bit/s per VN but the required bandwidth between physical servers increased to 1.0×10^9 bit/s per VN. The trade-off between the bandwidth loss and the required bandwidth was balanced by assigning every four VMs in the VN to a physical server, by which the required bandwidth of the outside racks was minimized (about 2.0×10^8 bit/s per VN). This solution is

coincident with a one-rack type of product offering for data centers; this product is delivered as a pre-configured single rack or multiple racks including physical servers, network switches, and virtualization software (e.g., [28]). This paper dealt with a single data center alone. The resource cost and performance would be different in an environment of multiple data centers and wide-area networks (WANs). In the future, we would therefore like to investigate VN allocation over WANs.

REFERENCES

[1] M. Mahalingam, D. G. Dutt, K. Duda, P. Agarwal, L. Kreeger, T. Sridhar, M. Bursell, and C. Wright, "VXLAN: A framework for overlaying virtualized layer 2 networks over layer 3 networks," http://tools.ietf.org/html/draft-mahalingam-dutt-dcops-vxlan-00, Aug. 2011.

[2] A. Fischer, J. Botero, M. Till Beck, H. de Meer, and X. Hesselbach, "Virtual network embedding: A survey," *IEEE Communications Surveys Tutorials*, vol. 15, no. 4, pp. 1888–1906, Feb. 2013.

[3] M. Rahman and R. Boutaba, "SVNE: Survivable virtual network embedding algorithms for network virtualization," *IEEE Transactions on Network and Service Management*, vol. 10, no. 2, pp. 105–118, Jun. 2013.

[4] C. Meixner, F. Dikbiyik, M. Tornatore, C. Chuah, and B. Mukherjee, "Disaster-resilient virtual-network mapping and adaptation in optical networks," in *Proc. of ONDM 2013*, Apr. 2013, pp. 107–112.

[5] H. Yu, V. Anand, C. Qiao, and G. Sun, "Cost efficient design of survivable virtual infrastructure to recover from facility node failures," in *Proc. of IEEE ICC 2011*, Jun. 2011, pp. 1–6.

[6] M. Habib, M. Tornatore, M. De Leenheer, F. Dikbiyik, and B. Mukherjee, "Design of disaster-resilient optical datacenter networks," *Journal of Lightwave Technology*, vol. 30, no. 16, pp. 2563–2573, Aug. 2012.

[7] P. Bodík, I. Menache, M. Chowdhury, P. Mani, D. A. Maltz, and I. Stoica, "Surviving failures in bandwidth-constrained datacenters," in *Proc. of the ACM SIGCOMM 2012*, Aug. 2012, pp. 431–442.

[8] J. Xu, J. Tang, K. Kwiat, W. Zhang, and G. Xue, "Survivable virtual infrastructure mapping in virtualized data centers," in *Proceedings of IEEE CLOUD 2012*, Jun. 2012, pp. 196–203.

[9] M. G. Rabbani, M. F. Zhani, and R. Boutaba, "On achieving high survivability in virtualized data centers," *IEICE Transactions on Communications*, vol. E97-B, no. 1, pp. 10–18, Jan. 2014.

[10] Q. Zhang, M. F. Zhani, M. Jabri, and R. Boutaba, "Venice: Reliable virtual data center embedding in clouds," in *Proceedings of IEEE INFOCOM 2014*, Apr. 2014.

[11] G. Hasegawa, T. Horie, and M. Murata, "Proactive recovery from multiple failures utilizing overlay networking technique," *Telecommunication Systems Journal*, vol. 52, no. 2, pp. 1001–1019, Feb. 2011.

[12] P. Bahl, R. Chandra, A. Greenberg, S. Kandula, D. A. Maltz, and M. Zhang, "Towards highly reliable enterprise network services via inference of multi-level dependencies," *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 4, pp. 13–24, Aug. 2007.

[13] R. Hinden, "Virtual router redundancy protocol (VRRP)," http://tools.ietf.org/html/rfc3768, Oct. 2012.

[14] VMware, Inc., "Protecting mission-critical workloads with VMware fault tolerance," http://www.vmware.com/files/pdf/resources/ft_virtualization_wp.pdf, Feb. 2009.

[15] P. Gill, N. Jain, and N. Nagappan, "Understanding network failures in data centers: Measurement, analysis, and implications," *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 4, pp. 350–361, Aug. 2011.

[16] R. Potharaju and N. Jain, "When the network crumbles: An empirical study of cloud network failures and their impact on services," in *Proc. of the ACM SOCC 2013*, Oct. 2013, pp. 15:1–15:17.

[17] X. Wu, D. Turner, C.-C. Chen, D. A. Maltz, X. Yang, L. Yuan, and M. Zhang, "NetPilot: automating datacenter network failure mitigation," in *Proc. of the ACM SIGCOMM 2012*, Aug. 2012, pp. 419–430.

[18] VMware, Inc., "VMware vSphere High Availability," http://www.vmware.com/products/datacenter-virtualization/vsphere/high-availability.html, Oct. 2012.

[19] M. Bari, R. Boutaba, R. Esteves, L. Granville, M. Podlesny, M. Rabbani, Q. Zhang, and M. Zhani, "Data center network virtualization: A survey," *IEEE Communications Surveys Tutorials*, vol. 15, no. 2, pp. 909–928, Second 2013.

[20] R. Jain, *The Art Of Computer Systems Performance Analysis*. John Wiley & Sons, Apr. 1991.

[21] D. G. Andersen, "Theoretical approaches to node assignment," *Computer Science Department*, pp. 1–12, Dec. 2002, http://repository.cmu.edu/compsci/86.

[22] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson, *Introduction to Algorithms*, 2nd ed. McGraw-Hill Higher Education, 2001.

[23] S. Luke, *Essentials of Metaheuristics*. Lulu, 2009, available at http://cs.gmu.edu/∼sean/book/metaheuristics/.

[24] W. E. Smith, K. S. Trivedi, L. A. Tomek, and J. Ackaret, "Availability analysis of blade server systems," *IBM Systems Journal*, vol. 47, no. 4, pp. 621–640, Oct. 2008.

[25] U. Hoelzle and L. A. Barroso, *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines*, 1st ed. Morgan and Claypool Publishers, 2009.

[26] D. S. Kim, F. Machida, and K. Trivedi, "Availability modeling and analysis of a virtualized system," in *Proc. of IEEE PRDC 2009*, Nov. 2009, pp. 365–371.

[27] K. Miettinen, *Nonlinear Multiobjective Optimization*. Springer US, Sep. 1998.

[28] Hitachi Data Systems, "Hitachi Unified Compute Platform Solution (UCP) Family," http://www.hds.com/, May 2014.