Abstraction Layer Based Service Clusters Providing Low Network Update Costs for Virtualized Data Centers

Ali Kashif Bashir, Yuichi Ohsita, and Masayuki Murata
Graduate School of Information Science and Technology
Osaka University
Osaka, Japan.
e-mail: {ali-b, y-ohsita, murata}@ist.osaka-u.ac.jp

***Abstract*— Network virtualization is one of the most promising technology for the data centers. It was innovated to use the network resources efficiently by evaluating new protocols and services on the same hardware. This paper presents a virtual distributed architecture for virtualized data centers. This architecture group virtual machines according to their service types. Further, it deploys an abstraction layer for each group that helps in managing, controlling, and maintaining the group. An abstraction layer and a group of virtual machines together form a cluster. This abstraction layer offers several advantages, however, in this work, we considered network update cost when recovering from failures as a parameter of evaluation. Simulation results prove that the proposed architecture can detect and replace failed machines (virtual and physical) with low costs in comparison with the centralized approaches.**

Keyword-network virtualization; service clusters; network failures; virtual data center network architecture; infrastructure for cloud applications; abstraction of virtual resources.

I. INTRODUCTION

This article is an extended version of our previous work [1]. It presents an architecture named Abstraction Layer based Service Clusters (AL-SC); in which virtual resources are grouped in clusters of various service types. Each cluster is managed and controlled by an Abstraction Layer (AL), which is a key element of this architecture and distinguishes it from the existing architectures. Deploying an AL offers several advantages to the virtual architecture; however, in this work we only evaluated the network update cost.

Network Virtualization (NV) [2] [3] [4] [5] [6] is one of the most promising technologies for the data centers (DCs). Introduced as a mean to evaluate new protocols and services [7]. It is already being actively used in research test-beds like G-Lab [8] or 4WARD [9], applied in distributed cloud computing environments [10]. Now, it is seen as a tool to overcome the obstacles of the current internet to fundamental changes. As such, NV can be thought of as an inherent component of the future inter architecture [11]. For DCs, it works as a backbone technology and let concurrent applications execute on a single hardware. Today, NV approaches are even applied in the telecommunication market, e.g., Open-Flow [12].

Virtualization is not a new concept. It is widely used to enhance the performance of DCs. With virtualization, we can create multiple logical Virtual Machines (VMs) on a single server to support multiple applications. These VMs takes the computation away from servers. VMware [13] and Xen [14] are two famous VMs proposed. However, virtualization of DC Networks (DCNs) aims at creating multiple Virtual Networks (VNs) at the top of a physical network [4]. Separation of VN/VNs from DCN offers several advantages, e.g., it allows to introduce customized network protocols and management policies. It lets concurrent applications run at the same underlying DCN and also help in securing the DC. On the other hand, without virtualization, we are limited to place a VM and also are limited in replacing or moving it.

VN, a primary entity in NV, is a combination of active and passive network elements (nodes and links) lies on top of a physical network. Virtual nodes are interconnected through virtual links, forming a virtual topology. With node and link virtualization, multiple VN topologies can be created and co-hosted on the same physical hardware. This virtualization introduces an abstraction that allows network operators to manage and modify networks in a highly flexible and dynamic way.

NV was envisioned to provide several features to the underlying infrastructure, e.g., scalability, flexibility, bandwidth improvement, etc. However, the existing virtual architectures provide only one or two features at a time. To enable the maximum features of NV for underlying architecture, in this paper, we propose AL-SC for DCs. The proposed architecture has two design aspects. First, it groups the VMs according to the service type they offer, e.g. VMs offering Map-reduce services can be grouped together. Note that, the number of services in an environment is defined by the network operator. Second, in order to manage, control, and monitor each groups, an AL is formed. It consists of a subset of VN switches that are separated with identifiers. A particular group of VMs and its' corresponding AL forms an SC. To the best of our knowledge, AL-SCs is the first proposed architecture that can be used in the service orchestration in the future.

An AL provides control to its cluster resources. Moreover, the number of switches in an AL are also variable. Due to these, ALs offer several advantages to the architecture of AL-SC like scalability, flexibility, better management and control, etc. We have discussed some of these features in our previous works [1] [15]. In this work, we evaluated the effectiveness of AL-SC in terms of network update cost such as when recovering from network failures, e.g., VM or server failures. Evaluation results prove that AL-SC requires low cost in comparison with the centralized virtual approaches. Though, we believe that AL-SC performs better than the existing distributed approaches also, e.g., adaptive VN [16]; however, in this work, we did not provide the comparison.

The rest of the paper is organized as follows: in Section II, we presented the background of FI model and related works of the paper. In Section III, we discuss the overview, topology and a few other concepts of AL-VC. Section IV includes the mechanism to construct ALs. Section V includes our evaluation and Section 6 concludes the paper.

II. RELATED WORKS

DCs have gained a significant attention and rapid growth in both scale and complexity and are acting as a backbone for the cloud applications [17]. Companies like Amazon EC. [18], Microsoft Azure [19], Facebook [20], and Yahoo [21] routinely use DCs for storage, search, and computations. In spite of their importance, architecture of today's DCs is far from being ideal due to following limitations:

- *No Performance Isolation:* Traditional DCs work as a single network and provide only best-effort solutions without performance isolation, which is required in modern cloud applications.

- *Inflexibility of the Network:* Due to non-flexible nature of traditional DCs, it is difficult to introduce new protocols or services. It leads to the minimal usage of the infrastructure.

- *Limited Management:* In the growing cloud application market, owners want control and need to manage the communication fabric for load-balancing, security, fault diagnosis, etc. However, the current architectures do not provide this flexibility.

- *Less Cost-effective:* Current DCs do not provide the support for multiple protocols. Applications usually require to migrate their management policies, VMs, and with the lack of support, the current infrastructure is not cost effective.

- *Internet Ossification* [22] [23]: The current Internet infrastructure is owned by a large number of providers, it is impossible to adopt a new architecture without the agreement of these stakeholders. Without consensus, any initiative to improve Internet services will be difficult in nature and limited in scope.

NV is seen as a solution to all these problems. Virtualization of DCs resolves above mentioned issues, on the other hand, they are required to manage the physical infrastructure in the best possible ways. In literature, several solutions proposed for purpose. We will discuss the most relevant ones in this work. In [13], the authors surveyed on the importance of virtualization to improve flexibility, scalability, and resource utilization for data center networks. Whereas, MobileFlow [24] introduces carrier-grade virtualization in EPC. Diverter [25] is a software based network virtualization approach that does not configure switches or routers. It logically partition IP networks for better accommodations of applications and services. VL2 [26] is a data center network architecture that aims at achieving flexibility in resource allocation. In VL2, all servers belonging to a tenant share a single addressing space regardless of their physical location meaning that any server can be assigned to any tenant.

SecondNet [27] focused on providing bandwidth guarantees among VMs in a multi-tenant virtualized DC. It assumes a VDC (Virtual Data Cluster) manager that created VDCs. This work achieves high scalability by moving information about bandwidth reservation from switches to hypervisors. It also allows resources to be dynamically allocated and removed from VDCs. Another VN architecture, CloudNaas [28] provides support for deploying and managing enterprise applications in the clouds. It relies on OpenFlow forwarding [12]. CloudNaas provides several techniques to reduce the number of entries required in each switch. CloudNaaS also supports online mechanisms for handling failures and changes in the network policy specification by re-provisioning the VDCs. In NetLord [29], a tenant wanting to run a Map-Reduce task might simply need a set of VMs that can communicate via TCP. On the other hand, a tenant running a three-tier Web application might need three different IP subnets, to provide isolation between tiers. Or a tenant might want to move VMs or entire applications from its own datacenter to the cloud, without needing to change the network addresses of the VMs.

PolyVine [30] and adaptive VN [16] are two more worth discussing distributed approaches. Polyvine embeds end to end VNs in decentralized manners. Instead of technical, it resolves the legal issues among infrastructure providers. In adaptive VNs [16], every server is supposed to have an agent. Each server agent communicates with another to make local decisions. This approach is expensive and needs additional hardware.

All the above mentioned approaches are usually application specific and discuss one objective at a time. There is hardly any approach that enable set of NV features to the underlying infrastructure. AL-SC tends to fill this wide and provides several features, some of them are discussed below.

- *Scalability and Flexibility:* Due to the distributed nature, SCs are easy to manage, monitor, and update. Each cluster can be managed independently without interrupting the operation of the network.

- *Facilitate service chaining:* Network Service Chaining (NSC) [31] [32], an emerging direction in NV, can be easily implemented on our service clusters.

- *Increase network administration control:* In Software Defined Networking (SDN) environments, where users need to have the control of the network to write the applications. Having ALs can help the network manager to hide underlying infrastructure.

- *Efficient Query Allocation:* Another advantage of service clustering is that it can save the search and allocation time of Virtual Network Requests (VNRs). When virtual resources are exclusively grouped in clusters, VNRs can quickly find their desired VMs.

Note that, some of these features of AL-SC are discussed in our previous works and some we plan to discuss in the future.

III. SYSTEM OVERVIEW

This section discusses the overview of AL-SC. This work does not contain any VN mapping algorithm. However, the algorithm can be adopted from the literature, e.g., [33] [34] [35]. In [33] a VN mapping algorithm is provided that maps the VNs to underlying physical network in distributed and efficient manners. In [34] VN mapping algorithm also meet the bandwidth demands. There are many other algorithms exists in the literature. Any of that can serve the purpose. Therefore, in this work, we assume that VMs are already mapped at the hosts. Table I includes the list of the abbreviations used in this paper.

TABLE I. USED ABBREVIATIONS

Acronym	Descriptions
AL-SC	Abstraction Layer based Service Clusters
AL	Abstraction Layer
NV	Network Virtualization
DC	Data Center
VM	Virtual Machine
DCN	Data Center Network
VN	Virtual Network
VM	Virtual Machine
SDN	Software Defined Networking
VNR	Virtual Network Request
SC	Service Cluster
SNS	Social Networking Service
NM	Network Manager
OPS	Optical Packet Switch
NSC	Network Service Chain

A. Architectual Overview

Service Clusters (SCs) are more desirable than physical DCs because the resource allocation to VC can be rapidly adjusted as users; requirements change with time [27]. In DCs, two servers providing similar service have high data correlation in comparison with servers providing different service [28]. This property is also reflected in their VMs. In other words, in order to execute one VNR, two machines (servers/VMs) offering similar services are likely to interact with each other more. Therefore, one motivation behind grouping VMs into SCs is to save the VNR allocation time. Logical representation of AL-SC is shown in Figure 1, where a DCN is virtualized into VCs of different service types, i.e., VC of Social Networking Services (SNSs), VC of Web services, VC of map-reduce, etc. This architecture can be implemented in several other ways. For example, in the environment where a single or multiple virtualized DCN are owned by multiple network operators. In that case, each operator can manage, control, and modify its own virtual resources in the shape of SCs. Classification of clusters according to the service or traffic type can be used in service orchestration in the future.

B. Topology

Ideally, VN topology should be constructed in a way that it achieves minimum energy consumption and larger bandwidth without delay. Minimum energy consumption can be achieved by minimizing the active number of ports and constructing energy efficient routes. Larger bandwidth can be achieved by adding virtual links in the VN and by managing traffic efficiently. Delay can be improved by using efficient routes and by processing data faster at switches. We argue that the proposed architecture has potential to provide all these features.

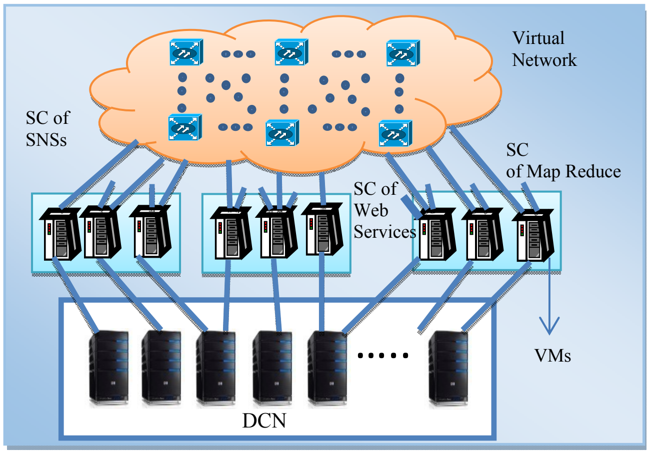


Figure 1. Overview of AL-SC.

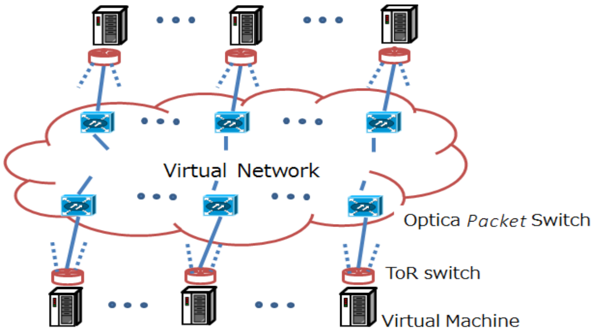


Figure 2. Construction of VN.

Connectivity of AL-SC is presented in Figure 2, where all the servers in a server rack are connected to one Top-of-the-Rack (TOR) switch. Each server is hosting multiple VMs. In the core of the network, to construct virtual links, we use Optical Packet Switches (OPSs). TOR switches produce electronic packets and they need to be converted into optical packets before sending over the optical domain of the network. Optical packets will be converted back to the electronic packets before forwarding to the TOR switches. This electronic/optical/electronic conversion is costly and should be reduced to increase the network performance. To read further on this, readers are suggested to read [36].

In AL-SC, every VM is connected to multiple OPSs. OPS that joins a particular AL can have four possible types of connections, namely: 1. with TOR switches, 2. with VMs of local cluster, 3. with OPSs of local AL, and 4. with OPSs of VN that are not part of its local AL. In Figure 3, we also presented the block diagram of this connectivity and as well the logical construction of AL-SC. After defining the topology, we would like to discuss the communication pattern in this architecture. Categorically, this communication is divide into two as:

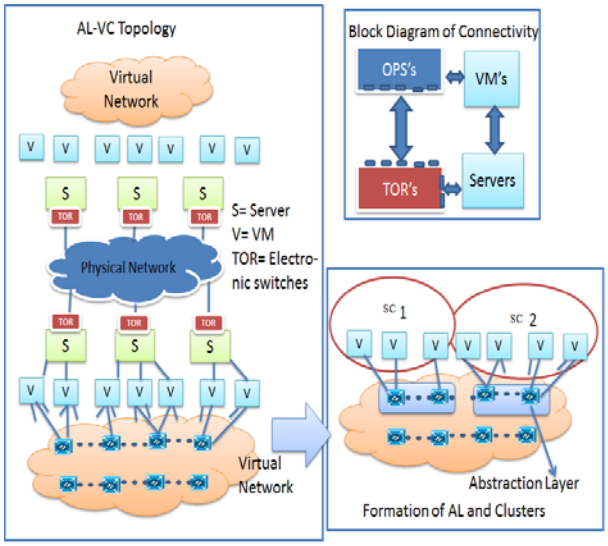


Figure 3. AL-SC Topology.

Intra-cluster Communication: Communication is intra-cluster when the destination VMs service type is the same as sender VM, which means it exists in the same SC. Most of the intra-cluster communication consists of small, but a large number of queries. AL- SC provides shorter routes to this traffic to reduce latency. Local switches of the corresponding AL of the particular cluster can interact with each other to find the destination VM, as shown in Figure 4.

Inter-cluster communication: Communication is inter-cluster when the destination VM belongs to another SC. This communication is usually less common than an intra-cluster. On the other hand, it generates a huge amount of traffic, hence, require high bandwidth, e.g., VM migration. Providing higher bandwidth is one of the characteristics of optical domain. Therefore, we construct optical paths in the VN as shown in Figure 4. We also have allocated a set of switches in an AL to handle inter-cluster traffic. Handling most of the queries (intra-cluster) within a cluster will motivate to providing dedicated paths for inter-cluster traffic.

C. *Optical Packet Switches*

Proposed topology can be constructed using packet switches. However, in order to achieve higher bandwidth with small energy consumption, we use OPS [37]. We modified the structure of OPS proposed by Urata et al. [38] as shown in Figure 5, where OPS is constructed of multiple wavelengths. Optical packets from other OPSs are de-multiplexed into optical signals of each wavelength. After label processing, these packets are relayed to the destination port. A shared buffer is constructed of CMOS. It stores the packets in case of collision or when packets received from

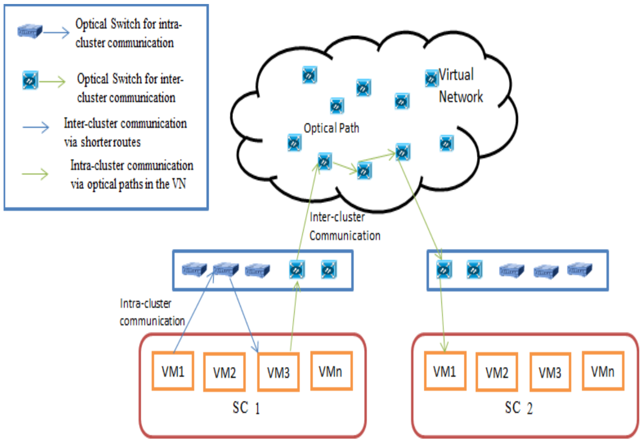


Figure 4. Communication in AL-SC.

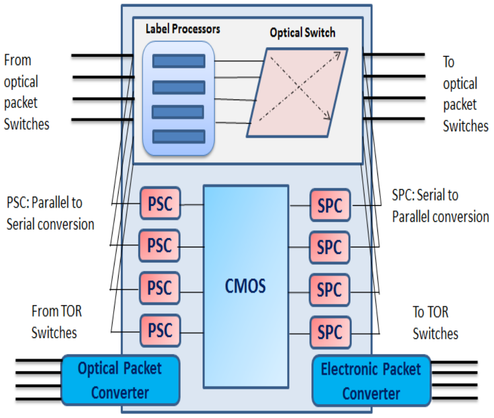


Figure 5. Structures of Optical Packet switch.

TOR switches. Packets from TOR switches are aggregated, stored in the buffer, and relayed to OPSs after parallel-to-serial conversion. Similarly, optical packets that need to be forwarded to TOR switches are converted to electronic packets. In simple words, this switch converts electronic packets to optical and vice versa. Detailed topology of AL-SC is explained in the next section.

D. *Network Manager (NM)*

A central NM is one of the most important entities, which is responsible for the operation of AL-SC. It decides the number of SCs, size of SCs, SC formation logic, and how they are mapped to the underlying DCN. Moreover, it also manages the physical resources like servers, VMs, links, etc. NM is responsible for VC formation and deletion. It also assigns each SC with a unique *SCID* and IP address. However, controlling and managing the cluster after creation is the job of its AL. In the future, ALs can be controlled by different application with the help of SDN. For address isolation, every SC has its own IP address space.

IV. ABSTRACTION LAYER

In the previous section, we presented the overview of AL-SC and highlighted its design aspects. In this section, we would like to present the AL construction algorithm.

A. *Construction of an AL*

The basic idea behind the construction of an AL is logically assigning a subset of the OPSs to a particular group of VMs. Group of VMs and an AL together is called a cluster in this work. In an AL, we assume every OPS knows the topology of its cluster, such as locations of VMs and their connections.

To construct an AL, VMs of every cluster selects the minimum subset of OPSs that connects them. This approach first selects the OPSs with highest connections and then OPSs with second highest connections and so on until all the VMs are connected. Finally, the subset of OPSs that covers all the VMs of a cluster will be declared as its AL. Switches of an AL will be differentiated from other OPSs of VN with the respective cluster ID. Information of these switches such as switch ID and IP addresses is forwarded to all the VMs. This procedure is repeated for every cluster until all the clusters have an AL.

Selecting switches with maximum connections reduce the number of switches in an AL, which will help in filtering and aggregating the traffic. On the other hand, it will increase the overhead at certain switches and will result in congestion. For that, we need to make sure that an AL has significant number of switches to handle congestion and to meet the required bandwidth demand. To meet these challenges, more refined algorithms are planned to be proposed in the future. The detailed mechanism of the current algorithm is as follows:

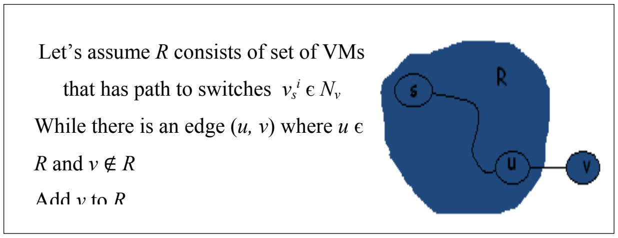
Step 1: As mentioned earlier, we assume that VMs are already mapped to servers and are grouped into clusters according to their service types. After this grouping, they connect themselves to the switches of VN. These connections can be established randomly or based on a specific criterion. In this work, we use random approach shown in Figure 6. The selection probability of the OPSs of AL is based on the distance, in which, we have

$$P_i = \frac{R_i}{\sum_j d_j} \tag{1}$$

Where
P_i = probability of selecting switches v_s^i
d_j = distance of switches from VM

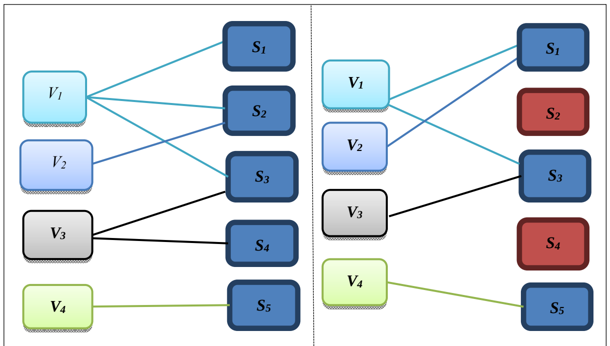
Step 2: Each VM sends a list of the OPSs they connect to the NM. Figure 7(a) shows this list.

Step 3: NM selects the minimum set of OPSs that cover all VMs of a group. To explain this, let's assume a graph $G = (V, E)$ with links $l_i \geq 0$, where the objective is to find a



Let's assume R consists of set of VMs
that has path to switches $v_s^i \in N_v$
While there is an edge (u, v) where $u \in R$ and $v \notin R$
Add v to R

Figure 6. Switch selection criteria.



(a) Candidate set
$V_1 = \{S_1, S_2, S_3\}$, $V_2 = \{S_1, S_2\}$, $V_3 = \{S_3, S_4\}$, and $V_4 = \{S_5\}$

(b) Final set
$AL = \{S_1, S_3, S_5\}$

Figure 7. Selection of an AL.

minimum subset of switches that covers all VMs. For this, we apply the following condition to VMs

$$S_i = \begin{cases} 0 & \text{if VM } v_s^i \text{ is not covered} \\ 1 & \text{if VM } v_s^i \text{ is covered} \end{cases}$$

Objective function: minimize $\Sigma 1 S$ for all v_s

Figure 7(b) is the final minimum subset of the OPSs required to form an AL for a cluster. These switches such as S_1, S_2, and S_3 in Figure 7(b), will be announced as an AL for a cluster. S represent an OPS. These OPSs will be assigned with SC_{ID}. In routing the traffic, OPSs in the intra-cluster phase can be addressed with (S_{ID}, and IP address) and in inter-cluster phase as (S_{ID}, SC_{ID}, and IP address).

Step 4: After selecting an AL, the remaining candidate switches will be discarded and they continue being part of the core of the VN.

This procedure is repeated until every cluster has an AL.

V. EVALUATION

NV plays a crucial role for DCNs. Since last few years, it is one of the most widely researched topics in cloud computing. Several architectures and solutions proposed that virtualize the physical resources. In all these works, underlying physical topology of the DCN plays an important role in the performance of virtual architectures as they provide the real grounds. Most of the existing virtual architectures are implementable on one or few topologies; however, AL-SC can be implementable on several topologies, such as B Cube, VL-2, FATTree, etc. It collects virtual resources from the underlying topology and group them according to the administrative logic as shown in the Figure 8. However, for evaluation of AL-SC, we choose FATTree [39] as underlying topology. In this work, we will evaluate the network update cost of AL-VC in terms of deletion or failure of VMs and in terms of addition of VMs. The cost is measured in the number of messages and time required. Therefore, we measured the communication cost in terms of messages and time when VMs are added or deleted in AL-SC architecture. Our evaluation environment is presented in Table Ⅱ.

TABLE II. ENVIRONMENT

Number of Servers	96
Number of VMs	360
Max VM a server can host	10
Number of switches in AL	10 % of VM in the cluster
Number of clusters	2, 4, 6, 8, and 10
DCN topology	FATREE
Parameters	Average time and Communication Cost

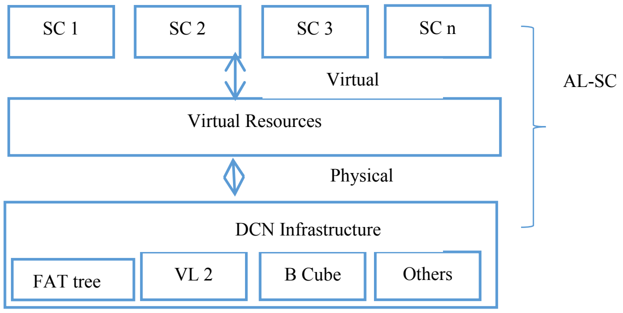


Figure 8. Implementability of AL-SC.

A. Network Update Cost in finding new VM

In this evaluation, we measured the time and communication cost in order to find a new VM. Several scenarios can be considered in this aspect. For example, migration of a VM from one server to another or from one cluster to another, failure or deletion of VM. In every scenario, our algorithms consist of following steps:

VM Discovery Mechanism

i. A VM is considered failed or migrated when it is not replying to the control messages of its AL.
ii. AL informs all the VMs of its cluster about this failure or movement with the ID of VM.
iii. AL will request the server whose VM is failed to host a new VM.
iv. If the server does not have enough resources to host a new VM, it will send attributes of the failed VM to the AL.
v. AL will request other servers that have the resources to host new VM.
vi. Servers will send the attributes of candidate VMs to AL.

vii. AL will select the server that has VM with the attributes most closed to fail VM.

viii. Finally, the failed VM will be replaced with a new VM.

The attributes of the requested VM can be represented as:

$$\alpha\tau\tau_{N\varpi} = ((\alpha\tau\tau_1, v_{\varpi}^{1}), (\alpha\tau\tau_2, v_{\varpi}^{2}), .., (\alpha\tau\tau_v, v_{\varpi}^{v})) \quad (2)$$

Non-functional (NF) attributes of the two VMs can be calculated by the following dissimilarity metrics.

$$dism(i,j) = \frac{\sum_{r=1}^{l} \delta_{ij}^{r} dis_{ij}^{r}}{\sum_{r=1}^{l} \delta_{ij}^{r}} \quad (3)$$

Where

l is the number of NF attributes

dis_{ij}^{r} denotes the dissimilarity of VM i and j related to $\alpha\tau\tau_{\lambda}$.

δ_{ij}^{r} expresses the coefficients of the *NF* attributes of machines i and j.

In Figures 9, 10, and 11, we evaluated the performance of AL-SC in detecting and replacing failed or migrated VM/VMs. To measure the performance of our algorithm, we constructed various cluster sizes in the network, e.g., 2, 4, 6, 8, and 10. Since we have no distributed control in the centralized approach, therefore, the detection of VMs will happen at the NM. For that, the central entity exchanges messages with all participating servers to discover a new server to host the new VM. However, in AL-SC this mechanism is executed within the cluster, i.e., on its AL. AL requires the fewer of messages and less time to find a new VM in comparison with centralized scheme.

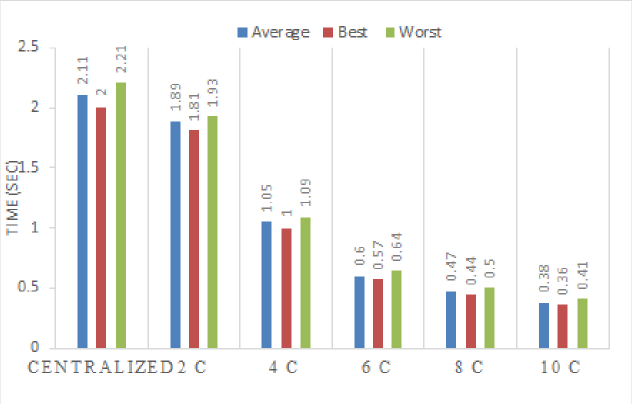


Figure 9. Time required to replace the failed VM.

In Figure 9, we measured the time required to replace a VM. In this evaluation, we consider three cases: the best, average, and the worst. In all three cases, we only measured the response time of machines in order to replace the VM. Whereas in Figure 10 represents the communication cost in terms of the number of messages required for this replacement. From these figures, we can see that an increased number of clusters decrease the average time and communication cost. This is because the number of participating entities in finding a new VM decrease. The increasing number of clusters helps in improving the performance of our algorithm. On the other hand, too many clusters may result in increased overhead, hence, a trade-off exists.

In Figure 11, we measure the time to replace the multiple failed VMs. The detection mechanism will remain same as mentioned above. It is clearly seen that performance of AL-SC significantly better than the centralized approach. Without SCs, NM has a lot of workload as a result of multiple failure detection and replacements. In case of AL-SC, each AL can run this mechanism locally the VM discovery procedure to find the new VMs with less overhead.

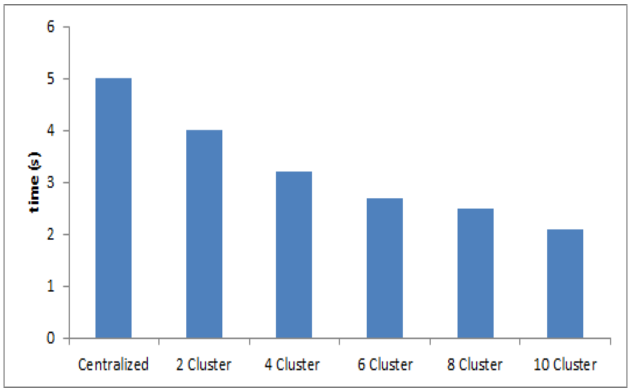


Figure 10. Communication Cost require in replacing the failed VM.

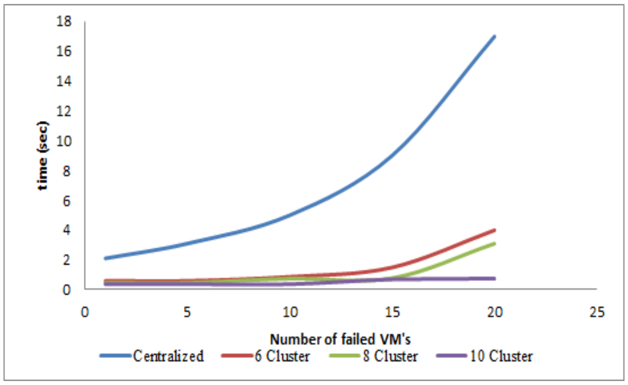


Figure 11. Average time required to replace the failed VMs.

B. Addition of VM

The architecture of AL-SC supports the addition of new VMs. In Figure 12, we evaluated the time required to add a new VM in the best, the average, and the ideal modes. The evaluation is conducted assuming no delay in the network. From the results, it is clear that if we have more refined clustering, i.e., more service types, it will help in saving the time required to add a VM. The algorithm will be as follows.

I. A VM request NM with a join message.
II. NM collects the NF attributes of the VM and matches it with the server attributes.
III. Server with the closest NF attributes will be requested to host the VM. If the server resources are limited, then the second closest NF server will be requested. This will continue until the host server is found.
IV. After joining a server, the VM will request to join the AL on the base of service type.
V. AL accepts the new VM and update its cluster topology and send the joining message with the ID of new VM to all existing VMs.

C. Network Update cost in terms of Server Failure

When a server fails, all the VMs hosted by that server will be considered failed. Therefore, to keep the operation of the SCs, new hosts for these VMs need to be found. The procedure of discovery will be as follows

I. If a VM does not respond to keep-alive messages, AL considers it failed and contact with its server.
II. If the server also does not respond, AL assumes that the server has failed.
III. As servers are physical resources of a DCN, an AL does not keep the attributes of the server.
IV. Therefore, it informs the NM and asks for the attributes of the failed server and its VMs. VMs attributes are stored in both the NM and in the hosting servers. However, server attributes can be fetched only from the NM.
V. After receiving NF attributes, AL runs a local VM discovery algorithm to find new hosts for the VMs as explained before.

In AL-SC, if the failed servers and VMs belong to only one cluster, it will not affect the operation of other clusters. To evaluate the update cost, we assume that the failing server was hosting three VMs that require new host now. The update cost of finding new host/hosts for these VMs is calculated in Figure 13 and 14, in which we can clearly see that the cost decreases as the number of cluster increases. In this evaluation, we assumed that all three VMs of the failing servers were belonged to one SC. Let us consider the case when VMs belongs to more than only one SC. In this situation, average time to find new host will remain same; however, the number of messages required, will increase as the VM attribute matching algorithm will run in the multiple SCs.

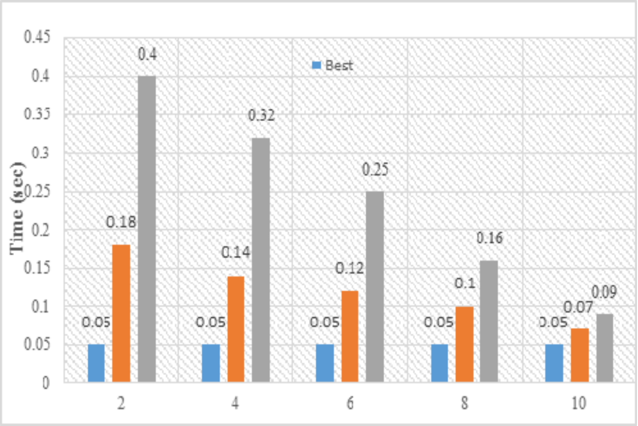


Figure 12. Time required to add a VM.

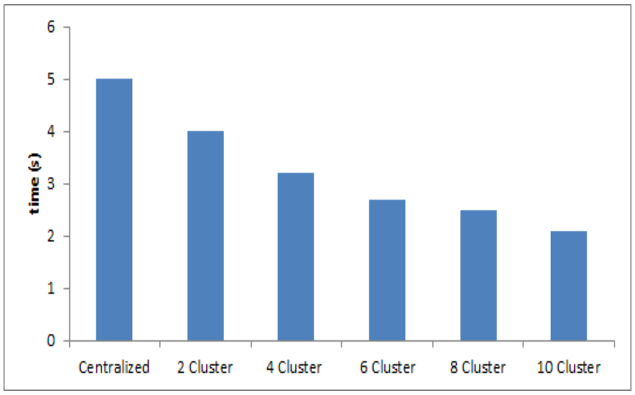


Figure 13. Average time to recover from a Server failure.

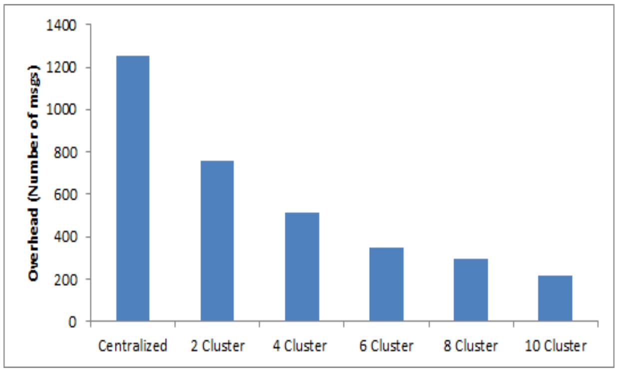


Figure 14. Communication cost require recovering from a server failure.

In case, the servers of a particular SC have no more resources to host new VMs, new servers will be requested from the NM. NM request the infrastructure provider to deploy new servers. They will be deployed in the infrastructure by joining the non-Functional attributes of the failing server. Attributes of the requested server can be represented as follows.

$$\alpha\tau\tau_{N\sigma} = ((\alpha\tau\tau_1,\ v_\sigma{}^1), (\alpha\tau\tau_2,\ v_\sigma{}^2), \ldots, (\alpha\tau\tau_v,\ v_\sigma{}^v)) \qquad (4)$$

VI. CONCLUSIONS AND FUTURE WORKS

The existing infrastructure of the data centers has several limitations. Network virtualization helps in overcoming these limitations and enables the cloud applications for users. In this paper, a distributed virtual architecture is proposed that virtualizes data center into multiple service clusters according to the service types exist in the network. Each service cluster consists of a group of VMs and a group of virtual switches called abstraction layer. An abstraction layer is the subset of virtual network switches and offers several features to the virtualized architecture. In this work, we only evaluated the network update cost when the virtual or the server machine fails. From the results, we can see that this architecture requires less time and cost in detecting the failures and recovering from them.

We plan to extend this work in the multiple directions. First, we will propose a more efficient mechanism for the construction of abstract layer. Second, we will improve other parameters, e.g., bandwidth with AL-SC. We also plan to adopt this architecture in the network service chaining.

ACKNOWLEDGEMENT

This work was supported by the National Institute of Information and Communications Technology (NICT), Japan.

REFERENCES

[1] A.K. Bashir, Y. Ohsita, and M. Murata, “Abstraction Layer Based Distributed Architecture for Virtualized Data Center,” in Proceedings of the 6th International Conference on Cloud Computing, Grids, and Virtualization (CLOUD COMPUTING 2015) IARIA, Mar. 2015, pp. 46-51, ISBN: 978-1-61208-388-9.

[2] N. M. M. K. Chowdhury, “Network virtualization: State of the art and Research Challenges,” Comm. Mag., IEEE, vol. 47, no. 7, pp. 20–26, Jul. 2009, doi: 10.1109/MCOM.2009.5183468.

[3] A. Berl, A. Fischer, and H. D. Meer, “Virtualisierung im Future Internet- Virtualisierungsmethoden Und anwendungen,” Info. Spek. Springer, vol. 33, no. 2, pp. 186–194, Apr. 2010, doi: 10.1007/s00287-010-0420-z

[4] K. Tutschku, T. Zinner, A. Nakao, and P. Tran-Gia, “Network Virtualization: Implementation steps towards the Future Internet,” Elec. Comm. EASST, vol. 17, Mar. 2009, doi: http://dx.doi.org/10.14279/tuj.eceasst.17.216

[5] D. Stezenbach, M. Hartman, and K. Tutschku, “Parameters and Challenges for Virtual Network Embedding in the Future Internet,” in Proceedings of Network Operations and Management Symposium (NOMS 2012) IEEE, pp. 1272-1278, Apr. 2012, ISSN: 1542-1201, ISBN: 978-1-4673-0267-8

[6] A. Berl, A. Fischer, and H. D. Meer, “Using System Virtualization to Create Virtualized Networks,” Elec. Comm. EASST, vol. 17, Mar. 2009, ISSN: 1863-2122.

[7] T. Anderson, L. Peterson, S. Shenker, and J. Turner, “Overcoming the Internet Impasse through Virtualization,” Comp. IEEE, vol. 38, no. 4, pp. 34–41, Apr. 2005, doi: http://doi.ieeecomputersociety.org/10.1109/MC.2005.136

[8] D. Schwerdel, D. Günther, R. Henjes, B. Reuther, and P. Müller, “German-lab Experimental Facility,” Third Future Internet Symposium (FIS 2010), vol. 6369, Sep. 2010, ISBN: 978-3-642-15876-6, doi: 10.1007/978-3-642-15877-3_1

[9] J. Carapina and J. Jiménez, “Network Virtualization: a view from the bottom,” in Proceedings of the Workshop on Virtualized Infra Syst. and Arch (VISA 09) ACM, pp. 73–80, 2009, ISBN: 978-1-60558-595-6, doi:10.1145/1592648.1592660

[10] P. Endo, A. Palhares, N. Pereira, G. Goncalves, D. Sadok, and J. Kelner, “Resource Allocation for Distributed Cloud: Concepts and Research Challenges,” Netw. IEEE, vol. 25, no. 4, pp. 42 –46, Jul. 2011, doi: 10.1109/MNET.2011.5958007

[11] N. Feamster, L. Gao, and J. Rexford, “How to lease the internet in your spare time,” Comp. Comm. Rev. ACM SIGCOMM, vol. 37, pp. 61-64, 2007, doi:10.1145/1198255.1198265

[12] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, and J. Rexford, “Openflow: Enabling Innovation in Campus Networks,” Comp. Comm. Rev. ACM SIGCOMM, vol. 38, no. 2, pp. 69–74, Apr. 2008, doi>10.1145/1355734.1355746

[13] M. F. Bari, R. Boutaba, R. Esteves, L.Z. Granville, M. Podlesny, and M.G. Rabbani, “Data Center Network Virtualization: A Survey,” Comm. Surv.& Tuto. IEEE, vol. 15, pp. 909-928, May. 2013, doi: 10.1109/SURV.2012.090512.00043

[14] WMware. “Virtualization Overview,” White paper, 2006. Available from: https://www.vmware.com/pdf/virtualization.pdf 2015.12.03

[15] A.K. Bashir, Y. Ohsita, and M. Murata, “A Distributed Virtual Data Center Network Architecture for the Future Internet,” Technical Reports of IEICE (IN2014-165), pp. 261-266, Mar. 2015.

[16] I. Houidi, W. Louati, D. Zeghlache, P. Papadimitriou, and L. Mathy, “Adaptive Virtual Network Provisioning,” in Proceedings of the 2nd Workshop on Virtualized Infrastructure Systems and Architectures (VISA 10) ACM SIGCOMM, pp. 41-48, Sep. 2010, ISBN: 978-1-4503-0199-2, doi: 10.1145/1851399.1851407

[17] P. Endo, A.D. A. Palhares, N. Pereira, and J. Mangs, “Resource Allocation for Distributed Cloud: Concepts and Research Challenges,” Netw. IEEE., vol. 25, no. 4, pp. 42 – 46, Jul. 2011, doi: 10.1109/MNET.2011.5958007

[18] “Amazon Web Services: Overview of Security Processes,” Amazon, pp. 1-75, Aug. 2015, Available from: https://d0.awsstatic.com/whitepapers/aws-security-whitepaper.pdf, 2015.12.03

[19] B. Calder, J. Wang, A. Ogus, N. Nilakantan, A. Skjolsvold, and S. McKelvie, “Windows Azure Storage: A Highly Available Cloud Storage Service with Strong Consistency,” 23rd Symposium on Operating System Principles (SOSP 11) ACM, pp. 143-157, 2011, ISBN:978-1-4503-0977-6 doi: 10.1145/2043556.2043571

[20] N. Farrington and A. Andreyev, "Facebook's Data Center Network Architecture," Facebook, Inc, Available from: http://nathanfarrington.com/papers/facebook-oic13.pdf, 2015.12.03

[21] Y. Chen, S. Jain, V. K. Adhikari, Z.L. Zhang, and K. Xu. "A First Look at Inter-Data Center Traffic Characteristics via Yahoo! Datasets," in Proceedings of the INFOCOM 2011, pp. 1620-1628, Apr. 2011, ISBN: 978-1-4244-9919-9, doi: 10.1109/INFCOM.2011.5934955

[22] J. S. Turner and D. E. Taylor, "Diversifying the Internet," in Proceedings of the Global Telecommunication Conference (GLOBECOM 05) IEEE, Dec. 2005, ISBN: 0-7803-9414-3, doi: 10.1109/GLOCOM.2005.1577741

[23] A. Bianzino, C. Chaudet, D. Rossi, and J. Rougier, "A Survey of Green Networking Research," Comm. Surv & Tut. IEEE, vol. 14, pp. 3-20, Feb. 2012, doi: 10.1109/SURV.2011.113010.00106

[24] K. Pentikousis, Y. Wang, and W. Hu, "MobileFlow: Toward Software-Defined Mobile Networks," Communications Magazine, IEEE, vol. 51, no. 7, pp. 44-53, July 2013

[25] A. Edwards, F. A, and A. Lain, "Diverter: A New Approach to Networking Within Virtualized Infrastructures," in Proceedings of the 1st Workshop on Research on Enterprise Networking (WREN 09) ACM, pp. 103-110, Aug. 2009, ISBN: 978-1-60558-443-0, doi: 10.1145/1592681.1592698

[26] A. Greenberg, J. Hamilton, N. Jain, S. Kandula, C. Kim, and P. Lahiri, "VL2: A Scalable and Flexible Data Center Network," in Proceedings of the Data Communication Conference (SIGCOMM) ACM, pp. 51-62, Oct. 2009, ISBN: 978-1-60558-594-9 doi: 10.1145/1592568.1592576

[27] C. Guo, G. Lu, J. H. Wang, S. Yang, C. Kong, and Y.Zhang. "SecondNet: A Data Center Network Virtualization Architecture with Bandwidth Guarantees," in Proceedings of the 6th International Conference (Co-NEXT 10), ACM, 2010, ISBN:978-1-4503-0448-1 doi: 10.1145/1921168.1921188

[28] T. Benson, A.A.A. Shaikh, and S. Sahu, "CloudNaaS: A Cloud Networking Platform for Enterprise Applications," in Proceedings of the Symposium on Cloud Computing (SOCC 11) ACM, Oct. 2011, ISBN: 978-1-4503-0976-9, doi: 10.1145/2038916.2038924

[29] J. Mudigonda, P. Yalagandula, J. Mogul, B. Stiekes, and Y. Pouffary, "NetLord: A Scalable Multi-Tenant Network Architecture for Virtualized Datacenters," in Proceedings of the SIGCOMM, ACM, pp. 62-73, Aug. 2011, ISBN: 978-1-4503-0797-0, doi:10.1145/2018436.2018444

[30] M. Chowdhury, F. Samuel, and R. Boutaba, "PolyViNE: Policy-based Virtual Network Embedding across Multiple Domains," in Proceedings of the Workshop on Virtualized Infrastructure Systems and Architectures (VISA 10) ACM SIGCOMM, pp. 49-56, Sep. 2010, ISBN: 978-1-4503-0199-2, doi: 10.1145/1851399.1851408

[31] M. Xia, M. Shirazipour, Y. Zhang, H. Green, and A. Takacs, "Optical Service Chaining for Network Function Virtualization Communication," Maga. IEEE, vol. 53, issue 4, pp. 152-158, Apr. 2015, doi: 10.1109/MCOM.2015.7081089J.

[32] M. Sanner, M. Ouzzif, and Y.H. Aoul, "DICES: a Dynamic Adaptive Service-driven SDN Architecture," in Proceedings of the Network Softwarization Conference (NetSoft) IEEE, pp. 1-5, Apr. 2015, doi: 10.1109/NETSOFT.2015.7116125

[33] J.F. Botero, X. Hesselbach, A. Fischer, and H.d. Meer, "Optimal Mapping of Virtual Networks with Hidden Hops," Tele. Syst. Springer, vol 51, no. 4, pp. 273-282, Dec. 2012, doi: 10.1007/s11235-011-9437-0

[34] I. Houidi, W. Louati, and D. Zeghlachie, "A Distributed Virtual Network Mapping Algorithm," in Proceedings of the International Conference on Communications (ICC 08) IEEE, pp. 5634-5640, May 2008, ISBN: 978-1-4244-2075-9, doi: 10.1109/ICC.2008.1056

[35] A. Fischer, J.F. Botero, M.T. Beck, H. de Meer, and X. Hesselbach, "Virtual Network Embedding: A Survey" Comm. Surv. & Tuto. IEEE, vol. 15, no. 4, pp. 1888-1906, Nov. 2013, doi: 10.1109/SURV.2013.013013.00155

[36] M. Xia, M. Shirazipour, Y. Zhang, H. Green, and A. Takacs, "Network Function Placement for NFV Chaining in Packet/Optical Data Centers," Light Wave Tech., vol. 33, no. 8, Apr. 2015.

[37] Y. Ohsita and M. Murata, "Data Center Network Topologies using Optical Packet Switches," in Proceedings of the 32nd International Conference on Distributed Computing Systems Workshops (ICDCSW) IEEE, pp. 57-64, Jun. 2012, ISBN: 978-1-4673-1423-7, doi: 10.1109/ICDCSW.2012.53

[38] R. Urata, T. Nakahara, H. Takenouchi, T. Segawa, H. Ishikawa, and R. Takahashi, "4x4 Optical Packet Switching of Asynchronous burst Optical Packets with a Prototype, 4x4 Label Processing and Switching Sub-system," Opt. Expr, vol. 18, no. 15, pp. 15283-15288, Jul. 2010, doi: 10.1364/OE.18.015283.

[39] M. Al-Fares, A. Loukissas, and A. Vahdat, "A Scalable, Commodity Data Center Network Architecture," in Proceedings of the Conference on Data Communication (SIGCOMM 08) ACM, pp. 63-74, 2008, ISBN: 978-1-60558-175-0, doi: 10.1145/1402958.1402967

[40] Y. Zhang, A.J. Su, and G. Jiang, "Evaluating the Impact of Data Center Network Architectures on Application Performance in Virtualized Environments," in Proceedings of the 18th International Workshop on Quality of Service (IWQoS) IEEE, pp. 1-5, Jun. 2010, ISBN: 978-1-4244-5987-2, doi: 10.1109/IWQoS.2010.5542728