FLAX: A Flexible Architecture for Large Scale Cloud Fabric

Yiyang Shao*, Yihang Luo*†, Xiaohe Hu*, Yibo Xue†, Yang Xiang‡, Kevin Yin§
* Department of Automation, Tsinghua University, Beijing, China
† Research Institute of Information Technology, Tsinghua University, Beijing, China
‡ Yunshan Networks Inc., Beijing, China
§ Chief Technology Office, Cisco China
{shaoyy11, hu-xh14}@mails.tsinghua.edu.cn, {yihangluo, yiboxue}@tsinghua.edu.cn
xiangyang@yunshan.net.cn, kyin@cisco.com

***Abstract*—The scalability and geographical location agility of data centers have become two key concerns for those critical cloud applications. However, it is still infeasible to build non-blocking data centers which are scalable, agile and cost-effective, given that current network devices are either closed high-end or performance limited, and the dedicated fiber is expensive and hard to expand. This paper proposes FLAX, a flexible architecture consolidating intra- and inter-cloud networks for large scale fabric. By leveraging on Software-Defined Networking techniques, FLAX can provide non-blocking application networks and scale out to millions of 10 gigabit ethernet ports across geographically-separated and arbitrarily-connected cloud data centers. Under the global view of network controllers, uniformed design of switches in different hierarchies and involving Wide Area Networks make it possible to fully use all network elements, and hence driving down the cost of network infrastructure. We present the architecture design and future work in this paper, and also a prototype deployed in one of the largest third-party data centers in eastern China.**

***Index Terms*—Software-Defined Networking; Cloud Fabric; Data Center Interconnection.**

I. Introduction

Modern networking techniques hasten the evolution of data centers to the cloud, giving us ability to turn the expansion from "scale up" to "scale out", reducing cost or even fundamentally subverting network infrastructures. As a result, cloud data centers become gradually bigger, and today's business becomes more distributed and mobile than ever [1]. Nowadays, many critical cloud applications are sensitive to geographical location agility, and often need local and remote disaster recovery at the same time. In order to provide universal application acceptability, data centers are needed to provide high performance as well as nonstop access, leading to a particular requirement of data center's intra-architecture design and geographical interconnection mechanism.

Many previous works focused on intra-architecture of data centers have been proposed to provide more scalable and flexible application services [2]–[4]. While the scalability has been intensively studied, the cost for building a data center was rarely involved. The Capital Expenditures (CAPEX) for building network infrastructure could constitute about up to 15% of the total data center expenditure [5]. This is significant, and hence driving down the cost of individual network elements is important and economic. Unifying all network elements allows for volume pricing on bulk purchases, which can remarkably reduce the CAPEX and maintenance cost. However, this is critical to the intra-architecture design. Another way is to introduce multiple network equipment vendors, which brings competitive pressures to drive costs down. A concomitant advantage of this method is the maximum flexibility of vendor equipment choices, which needs us to minimize the software feature requirements and use open standards.

Besides intra-architecture, elastic interconnections between non-blocking data centers in a large scale can be hard to implement in practice. Considering packet loss is unacceptable, Wide Area Networks (WANs) are typically forced to provide an up to 3 times bandwidth over-provisioning to face the peak traffic versus average [6], [7]. While this can easily handle the inevitable failure, it surfers a waste of infrastructures as well as operation and maintenance. Fortunately, the over-provisioning bandwidth provides us an opportunity to build interconnections of multiple cloud data centers over WANs. Leveraging Software Defined Networking (SDN) principles and OpenFlow, switches can be programmed controllable forwarding tables, and Traffic Engineering (TE) techniques can be adopted to guarantee a sustainably sufficient bandwidth to transport specific traffic, *i.e.,* data center traffic, over WANs. In this way, building geographical interconnections of multiple cloud data centers over WANs becomes realizable.

In this paper, we propose FLAX, a flexible architecture for large scale cloud fabric, aiming to provide high performance as well as nonstop access in cloud data centers. By uniforming network elements and leveraging on SDN techniques, FLAX carries the importance of data center's intra-architecture design and geographical interconnection mechanism, and has already been deployed in practice to operate cloud data centers.

Main contributions of this paper are as follows:

- *Flexible and scalable architecture* to compromise horizontal upgrades both intra- and inter-data centers.
- *Cost-efficient compatibility* for network infrastructures.
- *Practical deployment* of the prototype as a functional verification in a large data center in China.

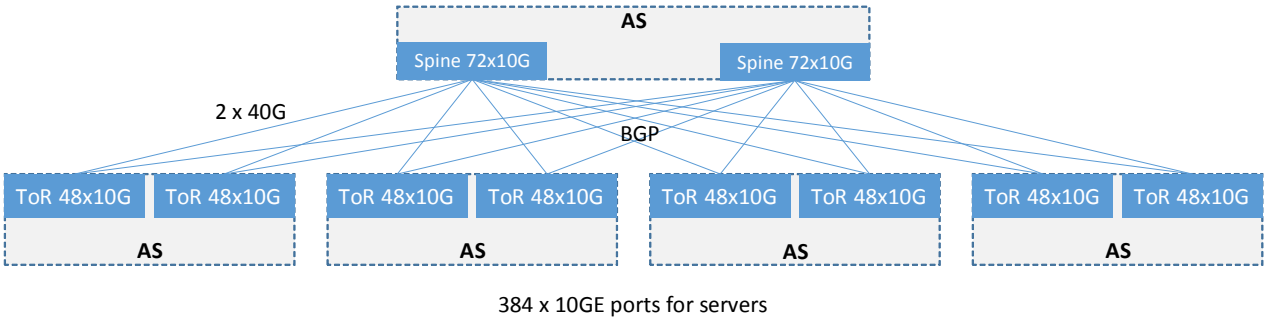

Fig. 2. A typical pod case with 2 spine switches and 8 ToR switches, which provides 384 10GE ports for servers

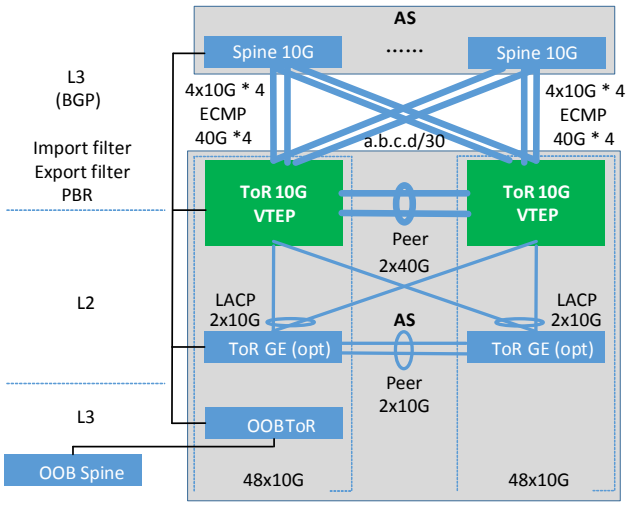

Fig. 1. Topology between ToR and spine switches

II. Architecture Design

In this section, we first show some key points which inspire our design. Then from basic pod design to interconnection mechanism, we demonstrate the FLAX architecture from the bottom up. Finally, we introduce the network controller.

A. Key points

Applications belong to different tenants are deployed in cloud data centers. In order to build a scalable virtual layer 2 network for tenant, we use VXLAN (Virtual eXtensible Local Area Network) to encapsulate heterogeneous traffic of different tenant, and built a scalable layer 3 cloud fabric using Border Gateway Protocol (BGP). Though in contrast with most traditional data center designs which use simple tree topologies and rely on extending layer 2 domains across multiple network devices, experimentation and extensive testing has shown that External BGP (EBGP) is well suited as a stand-alone routing protocol for large scale data center applications.

Until recently it was quite common to see the majority of traffic entering and leaving the data center, commonly referred to as "north-south" traffic. Traditional "tree" topologies were sufficient to accommodate such flows. However, today many large-scale data centers host applications generating significant amounts of server-to-server traffic, commonly referred to as "east-west" traffic. Scaling traditional tree topologies to match these bandwidth demands becomes either too expensive or impossible due to physical limitations, so we introduce new mechanism in this paper.

B. Topology of basic pod

Fig.1 is an illustration of the detailed topology between Top of Rack (ToR) switches and spine switches. In our design, each switch has two bonding Network Interface Cards (NICs) for Cloud service, and uses Link Aggregation Control Protocol (LACP) to double the bandwidth. In order to remove single point of failure, we connect every pair of ToR switches by two 40G peer links running peering protocol such as Multi Chassis Link Aggregation Group (MLAG). Each pair of ToR switches constitutes an Autonomous System (AS), and spine switches together constitute one. Besides, two Gigabit Ethernet (GE) switches are cross-linked to 10GE ToR switches. Finally, all the spine and ToR switches are connected by OOB (Out Of Band) ToR switches, and further connected by OOB spine switches. Through OOB switches, all the configuration of ToR and spine can be accomplished by a central network controller.

In our architecture, VXLAN VTEP (Virtual Tunnel End Point) is launched on ToR switches, and ToRs are connected by Layer 3 routing to each other. Fig.2 is the topology of a typical Cloud fabric, and we define it a "pod" in this paper. Every ToR switch connects to all spine switches using all 40GE links except for two peering links. Since a typical 10GE switch usually has forty eight 10GE ports and six 40GE ports which each can be split to four 10GE ports, we can use the same type of switches for both ToR and spine, achieving the uniformed design of network elements inside single pod.

C. Scaling basic pod

The pod above forms nuclear component of a data center. This original design makes its best to keep the coherence of different switches, and hence ensures diversity while scaling. If necessary, we can scale up by using devices with high port density, or scale out the pod topology by horizontally adding new switches. We show how to scale the basic pod design in the rest of this subsection.

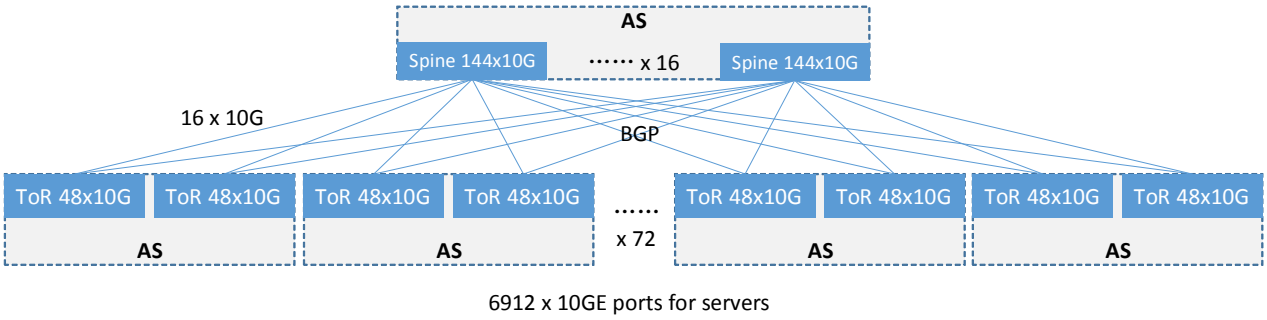

Fig. 3. A large pod case with 16 spine switches and 144 ToR switches, which provides 6912 10GE ports for severs

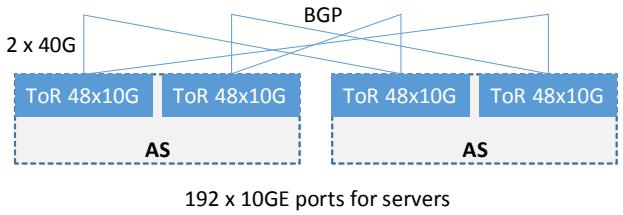

Fig. 4. A tiny pod case with 4 ToR switches, which provides 192 10GE ports for severs

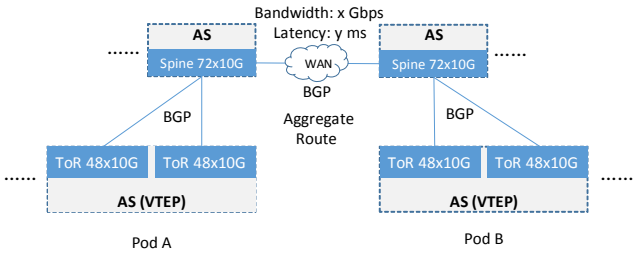

Fig. 5. Interconnection between two pods

Fig.3 illustrates a possible case for scaling up the topology. In this case we build the pod using 16 spine switches, each of them is 36×40GE chassis switch. Splitting each 40GE port to four 10GE ports, we can gain the spine layer with 36×4 10GE ports, so we are able to connect 144 ToR switches which can provide 6912 10GE ports if using the commonly-seen switches with 48×10GE ports and 6×40GE ports.

In some data centers, it also has elastic requirements for tiny pod. Fig.4 illustrates a simple case complying with such requirements. It only uses 4 ToR switches and can provide 192 10GE ports. Besides the large and tiny cases shown in this paper, leveraging FLAX architecture, infrastructure providers can pick up diverse topologies according to their demands.

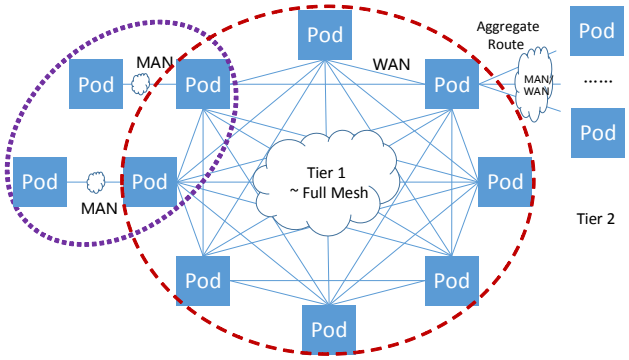

Fig. 6. Hierarchical Interconnecting

D. Hierarchical interconnection

We can further scale out the current 2-stage topology by adding a core layer to build a 3-stage Clos topology. However, it is expensive since core layer often needs extremely high performance switches. Instead of using more powerful spine switches or building more Clos stages, FLAX scales up using pod interconnecting. This mechanism brings some other advantages. Reminding that different pods should locate into different cities to allow local and remote diaster recovery, which are commonly essential conditions for today's data center, the design of hierarchical interconnection achieves geographical location agility at the greatest extent. What's more, the interconnection could be established directly over WANs, thus decreasing the expenditures.

In fact, pods usually already have WAN links, they can be naturally interconnected. What we concern is the bandwidth and latency of every WAN connection. Fig.5 illustrate the connecting between two pods. In order to make VTEPs in two pods reach each other, we need to assign a public IP address for every VTEP, or encapsulate private VXLAN by site-to-site Virtual Private Network (VPN).

Every pod in our design is not very large, but we have a large pod interconnecting network. As the scale grows, some pods will naturally form the Tier 1 of group, with luxuriant and high quality interconnecting links (large bandwidth or low latency). In ideal conditions, Tier 1 is composed of pods connected by a near full-mesh network, as Fig.6 illustrates.

Using interconnection mechanism, what we need to further design is the resource allocation algorithm of AS numbers for entire architecture. There are only 1023 private 2-bytes AS numbers can be used by switches, while this can be much more if using 4-bytes AS number. Besides, every pair of ToR

switches must have a VTEP IP address which is (directly or through VPN) reachable from other VTEPs. In order to reduce the routing table size, every pod should announce aggregate routes of its VTEPS, and Tier 1 should announce aggregate routes of directly connected Tier 2 pods.

E. Network controller

Besides configuration of network elements, including ToR and spine switches mentioned above, the main functions of controller are application placement and traffic engineering.

Given network conditions of the fabric, controller settles specific application into the data center based on its scale and demands. Network conditions here refer to static parameters such as inherent bandwidth and latency, and dynamic parameters which can be gained from monitoring. In case of network failure, controller will rearrange influenced applications. For those applications with backup, standby nodes will be settled as active ones, and new backups will be arranged.

Based on current conditions or even predictable future ones, traffic engineering can tune the traffic in fine-granularity according to applications' demands. This needs controller's powerful controllability over the entire network and deep realization of traffics, which sometimes need to leverage Deep Packet/Flow Inspection (DPI/DFI) techniques.

III. Deployment

Currently, the FLAX architecture and corresponding control system have been applied in industry. Suzhou International Science-park Data Center (SISDC for short), which is the largest third-party data center in eastern China and the first Tier IV Internet Data Center (IDC) in Asia, utilized FLAX to build up its cloud computing service.

For the detailed deployment, Centec Networks E350 has been utilized as the gigabit switch, Arista 7050 has been utilized as the 10 gigabit ethernet switch, and DELL R720 has been utilized as the computing node, which is used to virtualize as virtual machines. For the management scale, 10 racks are merged into 2 pods, one pod has 6 racks and the other has 4 racks. For the rack of virtual machines, Arista 7050 acts as the ToR switch, and for the rack of physical servers, Centec E350 acts as the ToR switch, beside of this, each rack can deploy 10 physical servers.

Each physical machine and virtual machine can be connected through L2 network under this deployment. With the design of FLAX, current deployment can be easily scaled out. Even more, when SISDC need to establish interconnection to another data center in the same city or a remote one, the network architects could achieve it directly using FLAX.

For the application scenario, SISDC's traditional business is hosting, only providing space rental and internet access service. With the deployment of FLAX, they expand their services to the interconnection and unified management with computing, storage, network device, etc., thus providing heterogeneous virtual private cloud services.

IV. Related Works

Folded Clos topology is a common choice for horizontally scalable data center topology, for the reason that it can be easily scaled out. This topology features an odd number of stages and is commonly made of uniform elements. Clos topology is fully non-blocking, or more accurately non-interfering. Using ECMP protocol, Clos topology is able to balance server-to-server traffic over all available paths.

Fat-tree [2] and VL2 [3] are the two typical data center network designs using commodity switches and non-blocking Clos topologies. Fat-tree proposes a customized routing primitive which is hard to be supported by existing switches. VL2 aims to realize multi-tenancy resource allocation with address mapping at ToR switches, while lack of absolute bandwidth guarantees between hosts. In contrast, FLAX offloads VXLAN to ToR switches to provide virtual L2 networks, and leverages classical ECMP and BGP protocols to meet bandwidth guarantees.

V. Conclusion and Future Works

Leveraging Software-Defined Networking techniques, we propose FLAX, a flexible architecture for large scale cloud fabric. We present the basic design of single pod, and then show how to scale it and hierarchically interconnect pods between data centers. Because of the uniform switch compatibility, it is possible to drive down expenditures of network infrastructure using FLAX. A prototype of our design has been deployed in SISDC to prove its efficiency .

As an ongoing work, we will mainly focus on two aspects in future. First, we only deploy a two-stage hierarchical interconnection in current version, and we will achieve multi-stage to scale out to millions of 10GE ports. Then, we will try to model diverse cloud business types based on their traffic demands, and let the topology accommodate each accordingly.

VI. Acknowledgments

This work was supported by the Tsinghua-CISCO Joint Research Center Foundation under Grant No.RFP2015-08.

References

[1] *Cisco Data Center Interconnection*. [Online]. Available: http://www.cisco.com/c/dam/en/us/solutions/collateral/data-center-virtualization/data-center-interconnect/at_a_glance_c45-493703.pdf

[2] M. Al-Fares, A. Loukissas, and A. Vahdat, "A Scalable, Commodity Data Center Network Architecture," in *ACM SIGCOMM*, 2008.

[3] 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 *ACM SIGCOMM*, 2009.

[4] X. Wang, Y. Qi, Z. Liu, and J. Li, "LiveCloud: A Lucid Orchestrator for Cloud Datacenters," in *IEEE CloudCom*, 2012.

[5] *Use of BGP for routing in large-scale data centers*. [Online]. Available: https://tools.ietf.org/html/draft-ietf-rtgwg-bgp-routing-large-dc-05

[6] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu *et al.*, "B4: Experience with a Globally-Deployed Software Defined WAN," in *ACM SIGCOMM*, 2013.

[7] C.-Y. Hong, S. Kandula, R. Mahajan, M. Zhang, V. Gill, M. Nanduri, and R. Wattenhofer, "Achieving High Utilization with Software-Driven WAN," in *ACM SIGCOMM*, 2013.