Building and Operating Distributed SDN-Cloud Testbed with Hyper-Convergent SmartX Boxes

Aris Cahyadi Risdianto, Junsik Shin, and JongWon Kim(✉)

Gwangju Institute of Science and Technology (GIST), Gwangju, South Korea
{aris,jsshin,jongwon}@nm.gist.ac.kr

Abstract. In this paper, we describe our efforts for building and operating a distributed SDN-Cloud testbed by utilizing hyper-convergent SmartX Boxes that are distributed across multiple sites. Each SmartX Box consists of several virtualized functions that are categorized into SDN and cloud functions. Multiple SmartX Boxes are deployed and inter-connected to build a multi-regional distributed cloud infrastructure. The resulting deployment integrates both cloud multi-tenancy and SDN-based slicing, which allow developers to run experiments in a distributed SDN-Cloud testbed. It also offers enhanced troubleshooting capability by providing semi-automated resource configuration and tapping functionality.

Keywords: Hyper-convergent SmartX Box · Distributed and virtualized cloud · Software-defined networking · Future Internet testbed · DevOps automation

1 Introduction

Motivated by worldwide Future Internet testbed deployments (e.g., GENI [1], FIRE [2]), OF@TEIN project is started to build an OpenFlow SDN-enabled testbed over TEIN infrastructure in 2012 [3]. Several experimentation tools are developed to support both developers and operators in utilizing OF@TEIN testbed. Initially, a mixed combination of tools, ranging from simple web-/script-based to DevOps (Development and Operation) Chef-based automated tools, are deployed over SmartX Racks [4]. SmartX Rack consists of four devices: Management and Worker node, Capsulator node, OpenFlow switch, and Remote power device. Physically LAN-connected SmartX Racks are interconnected by L2 (layer 2) tunnels, employed in Capsulator nodes. However, SmartX Racks with multiple devices are subject to physical remote reconfigurations, which are extremely hard to manage for distributed OF@TEIN testbed. Thus, from late 2013, a hyper-convergent SmartX Box is introduced to virtualize and merge the functionalities of four devices into a single box. In 2015, multiple hyper-convergent SmartX Boxes are deployed over 9 countries, as shown in Fig. 1.

SDN-based tools can assist developers and operators to prepare testing environment over OF@TEIN infrastructure, by enabling networking resources (e.g., switches and Flowspaces) preparation. Similarly, cloud management software is ready to cover computing resource (e.g., VMs) preparation. The combination for distributed SDN-Cloud testbed can provide scalable and flexible computing resources with enhanced

Y. Zhang et al. (Eds.): CloudComp 2015, LNICST 167, pp. 224–233, 2016.
DOI: 10.1007/978-3-319-38904-2_23

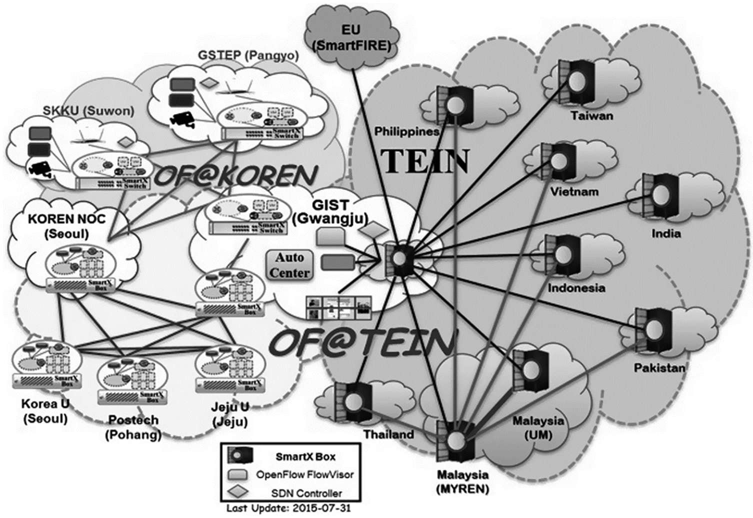

Fig. 1. OF@TEIN infrastructure.

networking capability. Thus, the seamless integration of SDN-enabled and cloud-leveraged infrastructure is a very challenging task, due to open and conflicting options in configuring and customizing resource pools together. That is, we should carefully provision all resource configuration aspects such as regional partitioning, virtualized resource slicing (i.e., isolation), and multi-tenancy support while considering hardware deployment for SDN-Cloud integrated.

As mentioned above, this paper introduces the hyper-convergent SmartX Boxes that can easily accommodate virtualized and programmable resources (i.e., OpenFlow-enabled virtual switches and OpenStack-leveraged cloud VMs) to build the OF@TEIN SDN-Cloud testbed. The collection of OpenFlow-enabled virtual switches is providing SDN capability, to-be-controlled by both developers and operators SDN controllers. At the same time, OpenStack-leveraged VMs (i.e., working as virtual Boxes) can be effectively managed by the OpenStack cloud management (i.e., OpenStack Keystone, Horizon, Nova, Glance, and others). The main contributions of this paper are:

1. Deploying OpenStack-leveraged computing/storage resources on top of SDN-enabled programmable networking resources for both developers and operators.
2. Providing OpenStack-leveraged multi-tenancy management interfaces as well as sliced network programming for multiple developers in multi-region distributed SDN-Cloud deployment.
3. Enhancing the semi-automated DevOps operation of OF@TEIN testbed by leveraging cloud deployment tools and OpenFlow-based SDN configuration tools with scaled-out hyper-convergent SmartX Boxes.
4. Prototyping and verifying the basic capability for resource-/flow-level visibility about hyper-convergent SmartX Boxes and their inter-connections.

2 Design of Distributed SDN-Cloud Testbed

There are several design aspects for distributed SDN-Cloud deployment: hyper-convergent SmartX Boxes, virtualized functions inside SmartX Boxes, automated provisioning tools for SmartX Boxes, and orchestrated control/management of the virtualize functions for SDN-Cloud integrated services.

2.1 Computing/Networking-Balanced Testbeds

It is well known that service-centric networking model to provide higher-level connectivity and policy abstraction is an integral part of cloud-leveraged applications. The emerging SDN paradigm can provide new opportunity to integrate cloud-leveraged services with enhanced networking capability through deeply-programmable interfaces and DevOps-style automation. A number of SDN solutions have been proposed to provide virtualized overlay networking for multi-tenancy cloud infrastructure. For example, Meridian has developed a SDN controller platform to support service-level networking for cloud infrastructure [5]. Also, CNG (Cloud Networking Gateway) attempts to address multi-tenancy networking for distributed cloud resources from multiple providers while providing flexibilities in deploying, configuring, and instantiating cloud networking services [6].

The large-scale deployment of GENI Racks over national R&E (research and education) backbone is also moving towards a programmable, virtualized, and distributed collection of networking/compute/storage resources, a global-scale "deeply programmable cloud". It will satisfy research needs in wide variety of areas including cloud-based applications. Moreover, GENI participates in significant international federations such as Trans-Cloud and the "Slice Around the world" efforts that include production (e.g., Amazon web services) cloud computing services for federations [1]. Another effort from EU, known as "BonFIRE" is a multi-site testbed that supports testing of cloud-based distributed applications, which offer a unique ease-to-use functionality in terms of configuration, visibility, and control of advanced cloud features for experimentation [7].

From these previous works for SDN-Cloud integration, there are several key requirements in integrating cloud services with SDN programmable infrastructure.

(1) The basic requirement is to provide cloud providers and tenant users to control and manage their own applications (functions for service chaining) as well as the connectivity among their applications on distributed cloud services.
(2) The connectivity requirements include the per-tenant creation and construction of virtual topology and network-layer information (i.e. switches, routers, subnets and access control lists).
(3) The fine-grained control over networking paths between distributed cloud services is required to provide fast failover and traffic prioritization, which utilizes OpenFlow-enabled SDN networking capabilities.
(4) The integration requirement to couple different APIs (application programming interfaces) is also needed to harmonize cloud-based resource/service orchestration and SDN-based virtual networking controllers.

2.2 Hyper-Convergent SmartX Box: Design

As mentioned above, the deployment conversion from SmartX Racks to SmartX Boxes are completed to better manage distributed cloud-leveraged services on the top of SDN-enabled inter-connect capabilities [8]. Thus, the design of hyper-convergent SmartX Boxes needs to carefully consider and balance both cloud and SDN aspects. The open-source OpenStack [9] cloud management software provides VM instances and basic networking options for diverse tenants. For SDN, several instances of virtual switches (derived from open-source Open vSwitch [10]) are installed and configured while allowing users/developers to share them simultaneously. Note that, to support the flexible remote configuration, each hyper-convergent SmartX Box requires dedicated P/M/C/D (power, management, control, and data) connections, which will be explained later [8].

However, besides the flexible dedicated connections, there are no specific hardware requirements for hyper-convergent SmartX Boxes. Therefore any commodity hardware with reasonable computing, storage, and networking resources can be utilized. The total amount of hardware resources will affect the capacity (e.g., total number of VM instances per flavor types) in specific boxes, sites, and regions. It is also important to consider the hardware acceleration support for virtualization and networking.

By merging all the required functionalities into the hyper-convergent SmartX Box, it is easier to realize the scale-out capability of SDN-Cloud testbed by simply adding hyper-convergent SmartX Boxes to increase the resource capacity of OF@TEIN infrastructure.

2.3 Virtualized SDN-Enabled Switches and Cloud-Leveraged VMs

In order to provide the integration of SDN-Cloud functions inside a single hyper-convergent SmartX Box, we arrange relevant functions as shown in Fig. 2. SDN-related virtual functions consist of several virtual switches with different roles, e.g., creating developer's networking topology, inter-connecting OpenFlow-based overlay networking, and tapping flows for troubleshooting. Also, cloud-related functions are placed to include VM instances for cloud-based applications and to support external connections to VMs.

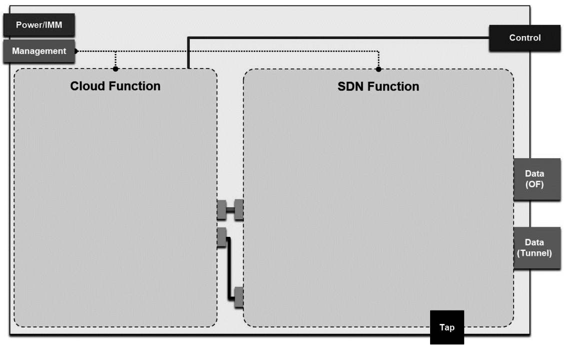

Fig. 2. Hyper-convergent SmartX Box: SDN-Cloud integration (top-level view).

The inside view of SDN-/Cloud-related functions are depicted in Fig. 3. First, SDN virtual switches are placed to cover named functionalities: *brcap* for capsulator (encapsulate OpenFlow packets through an overlay tunnel), *br1* and *br2* for users/developers switches, and *brtap* for tapping (capturing packets for troubleshooting [11]). Cloud-related VM instances (a.k.a., virtual Boxes: vBoxes) are managed by KVM hypervisors, which is controlled by OpenStack *Nova* with specific flavors and images. Additionally, virtual switches (i.e., *br-int, br-ex,* and *br-vlan*) and user-space virtual router are configured by OpenStack *Neutron* to provide required connectivity to cloud VM instances.

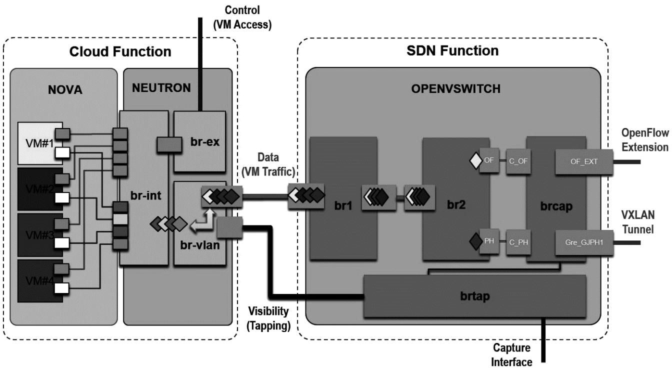

Fig. 3. Hyper-convergent SmartX Box: cloud-related (left) and SDN-related (right) functions.

2.4 Automated Provisioning

Deploying hyper-convergent SmartX Boxes in heterogeneous physical (i.e., network topology) environments is quite challenging, since it is subject to different performance parameters and independent network administrative domains. As a consequence, it is very hard to sustain the operation of all hyper-convergent SmartX Boxes. Thus, a set of automated provisioning tools is developed to minimize the consumed time for installing and configuring hyper-convergent SmartX Boxes with pre-arranged P/M/C/D connections. The P (Power) connection is used for power up/down SmartX Boxes. The M (Management) connection is mainly used for managing SmartX Box as the operator. Also, the C (Control) connection is used to access and control the SDN-/Cloud-related functions (i.e., virtual switches and VMs). Finally, the D (Data) connection is used for any kind of data-plane traffic that includes inter-connection traffic among multiple SmartX Boxes. Also, the automated provisioning tools are controlled by a centralized Coordinator Box, which has full access to all distributed hyper-convergent SmartX Boxes.

First, in order to automate SDN vSwitches provisioning, *ovs-vsctl* high-level programming interface for Open vSwitch is utilized. Note that *ovsdb* (Open vSwitch database) protocol is also utilized for the centralized configuration of Open vSwitch database inside each hyper-convergent SmartX Box. The provisioning task includes the creation of virtual switches, the configuration of virtual ports/links and overlay tunnels, and the connection of virtual switches and SDN controllers. Next, open-source OpenStack cloud software has special installation and configuration tools, called as *"DevStack"* that can support several modes of OpenStack configurations with selected OS (e.g., Ubuntu, Redhat Enterprise Linux, and CentOS) [12]. For OF@TEIN SDN-Cloud testbed, we customize DevStack-based installation and configuration to facilitate multi-regional OpenStack cloud deployment with centralized management and authentication.

2.5 SDN-Cloud Centralized Management

SDN-/Cloud-related virtual functions are inter-connected to provide the end-to-end communication for cloud-leveraged applications. The Cloud-related virtual functions are managed centrally by open-source OpenStack cloud management and orchestration software [9]. The SDN-related virtual functions are also centrally controlled by the ODL (Open Daylight) SDN Controller [13]. OpenStack *Keystone* is used for centralized user authentication. OpenStack *Nova* and OpenStack *Neutron* is utilized to create VM instances and to provide enhanced connectivity, respectively. The ODL SDN controller manipulates the flowtable entries of SDN-enabled virtual switches to enable the flexible steering of inter-connection flows among various functions located in different cloud regions (sites). Both cloud management software and SDN control software are required to mix and match the configurations so that we can ensure the consistent connections between cloud VM instances. Remember that the main challenge is how to accommodate cloud-based multi-tenancy virtual networks (e.g., flat, VLAN, or tunneled network) for OpenFlow-based network slicing (e.g., IP subnets, VLAN IDs, and TCP/UDP ports). Eventually VLAN-based multi-tenancy traffic control (e.g., tagging, steering, and mapping) is chosen to integrate tenant-based and slicing-based SDN-Cloud networking.

3 Verification and Evaluation

We now explain the details of SDN-Cloud testbed deployment, verification steps, and preliminary measurement results.

3.1 Multi-Site Deployment for Distributed SDN-Cloud Testbed

As shown in Fig. 4, multiple hyper-convergent SmartX Boxes are deployed for OF@TEIN testbed. Current deployment is focusing on migrating from limited open-source XEN VM hypervisor to open-source OpenStack cloud management. The OF@TEIN testbed relies on the heterogeneous physical underlay networks across multiple administrative domains. Thus, the multi-regional OpenStack cloud deployment is currently investigated as the deployment option, because it gives simple and common configuration for all regions (i.e., SmartX Box sites), supports independent IP addressing scheme, and has less dependency on the overlay networking among regions. Despite of multi-regional independent cloud deployment, the OF@TEIN testbed supports an integrated cloud management interface by deploying web-based OpenStack *Horizon* UIs and the centralized account/token authentication from OpenStack *Keystone*. The resulting OpenStack multi-regional cloud deployment is illustrated in Fig. 5 for OF@TEIN testbed.

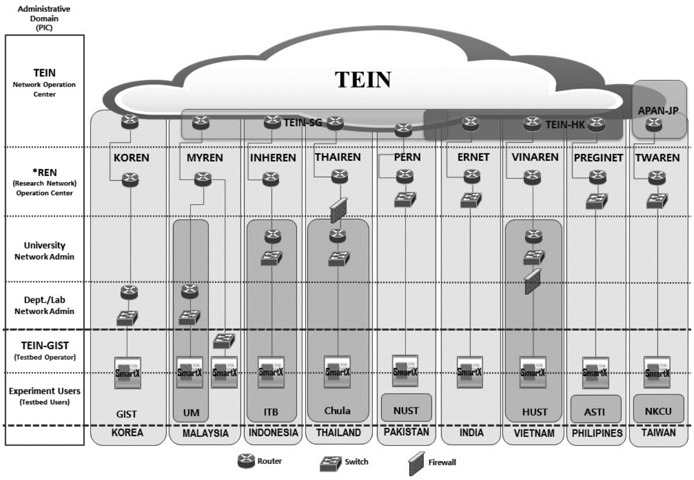

Fig. 4. OF@TEIN deployment over multi-domain R&E networks.

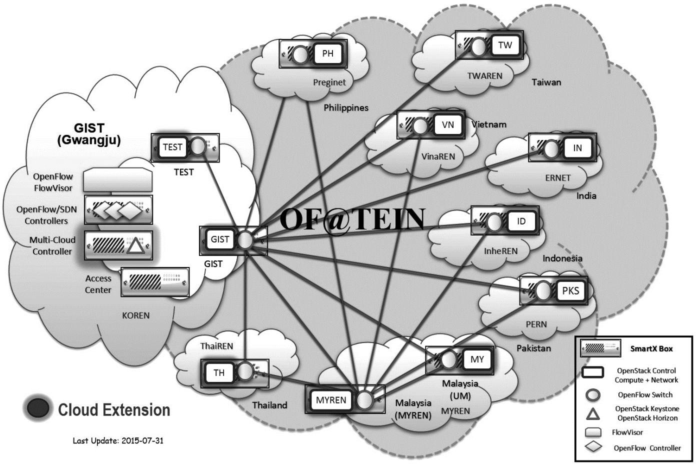

Fig. 5. OF@TEIN multi-regional cloud deployment.

Next, the OF@TEIN testbed is enhanced with multiple mesh-style inter-connections of NVGRE/VXLAN overlay tunnels, along with the special flow-tapping virtual switch [11]. The OpenStack multi-region deployment is modified to build a SDN-Cloud integrated testbed where inter-VM connectivity between cloud VMs is utilized by leveraging OpenFlow-enabled data planes. The data planes are programmed and controlled by the centralized SDN controller, co-located with the centralized cloud management.

3.2 Semi-Automated Provision: Verifications and Evaluations

In order to facilitate the agile deployment of OF@TEIN testbed, both SDN-/Cloud-related tools are utilized for automated provisioning of hyper-convergent SmartX Boxes. This is aligned with the recent employment of DevOps automation, since the OF@TEIN testbed is operated by a limited number of operators and becomes easily uncontrollable as it spans across multi-domain inter-connected networks beyond the privileges of testbed operators. Thus, by utilizing *DevStack*-based OpenStack deployment and *ovs-vsctl* or *ovsdb* protocol for virtual switch provisioning, we can simplify the semi-automated provisioning of hyper-convergent SmartX Boxes. In addition, REST APIs of ODL SDN controller is utilized for automated flow insertion, flow modification, and flow deletion. In summary, most of provisioning steps are automated with the exception of manual handling of critical tasks such as *DevStack*-based OpenStack service restart and VXLAN tunnel checking/recovery.

The semi-automated deployment shows takes around 50 min for fully upgrading a SmartX Box with ~300 Mbps network connection. It takes around 6 h for the slowest network connection of <10 Mbps. However, it takes only around 20 min (including Box restart) for installation and configuration without cleaning up the previous installation and upgrading the operating system. Moreover, it takes less than 10 min to re-configure hyper-convergent SmartX Box with offline mode (i.e., no online copy from OpenStack repository) (Table 1).

Table 1. Provisioning time comparison of sites with different network connection speed.

Site ID	Network connection speed (Mbps)	Installation time
GIST	~ 200	50 min
MY	~ 100	4 h
PH	~ 10	6 h

3.3 Example Experiments with SDN-Cloud Playground

By manipulating both OpenStack cloud management and OpenFlow-enabled SDN control, as depicted in Fig. 6, an experiment for deploying VLAN-based multi-tenancy traffic control is designed to verify the SDN-Cloud integration. First, we place VMs in two cloud regions and prepare the connectivity for these VMs. These VMs are tagged by OpenStack *Nova* with specific tag ID. Second, OpenStack *Neutron* automatically maps and matches VLAN IDs with SDN-based slice parameters. This allows inter-connection flows for VMs to be steered by the developer's SDN controller, supervised by FlowVisor [14]. The SDN-based flow steering inserts flowtable entries according to the particular incoming and outgoing ports in developer's virtual switches, where several ports are mapped to other cloud regions/sites. Finally, based on the destination site, it maps to a specific tunnel interface that is pre-configured by the SDN controller of operators. Eventually, the testing of end-to-end connections between VMs is required to verify the consistency of flow tagging, steering, and mapping for specific testing flows.

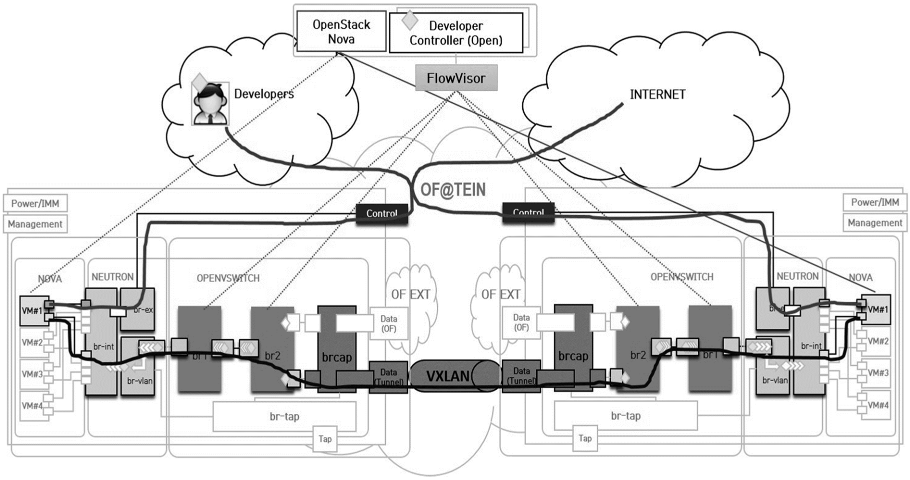

Fig. 6. An experiment example over OF@TEIN SDN-Cloud testbed.

Acknowledgements. This work makes use of results produced by the SmartFIRE project, which is supported by the International Research and Development Program of the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT and Future Planning (MSIP, Korea) (Grant number: K2013078191) and the Seventh Framework Programme (FP7) funded by the European Commission (Grant number: 611165).

References

1. Berman, M., et al.: GENI: A federated testbed for innovative network experiments. Comput. Netw. **61**, 5–23 (2014). doi:10.1016/j.bjp.2013.12.037
2. FIRE: Future Internet Research and Experimentation, September 2015. http://www.ict-fire.eu/home.html
3. Risdianto, A.C., Na, T., Kim, J.: Running lifecycle experiments over SDN-enabled OF@TEIN testbed. In: Proceedings of the IEEE International Conference on Communications and Electronics, July–August 2014
4. Risdianto, A.C., Kim, J.: Prototyping media distribution experiments over OF@TEIN SDN-enabled testbed. In: Proceedings of the 40th Meeting of the Asia-Pacific Advanced Network, vol. 38, pp. 12–18 (2014). http://dx.doi.org/10.7125/APAN.38.2
5. Banikazemi, M., Olshefski, D., Shaikh, A., Tracey, J., Wang, G.: Meridian: an SDN platform for cloud network services. IEEE Commun. Mag. **51**(2), 120–127 (2013)
6. Mechtri, M., Houidi, I., Louati, W., Zeghlache, D.: SDN for inter cloud networking. In: IEEE SDN for Future Networks and Services (SDN4FNS), Trento, Italy, November 2013
7. Kavoussanakis, K., et al.: BonFIRE: The clouds and services testbed. In: IEEE 5th International Conference on Cloud Computing Technology and Science (CloudCom), vol. 2, pp. 321–326, December 2013
8. Risdianto, A.C., Shin, J., Ling, T.C., Kim, J.: Leveraging ONOS SDN controllers for OF@TEIN SD-WAN experiments. In: Proceedings of the 40th Meeting of the Asia-Pacific Advanced Network, vol. 40, pp. 1–6 (2015). http://dx.doi.org/10.7125/40.1

9. Sefraoui, O., Aissaoui, M., Eleuldj, M.: OpenStack: toward an open-source solution for cloud computing. Int. J. Comput. Appl. **55**(3), 38–42 (2012)
10. Pfaff, B., Pettit, J., Amidon, K., Casado, M., Koponen, T., Shenker, S.: Extending networking into the virtualization Layer. In: Proceedings of the HotNets, Chicago, IL, October 2009
11. Risdianto, A.C., Kim, J.: Flow-centric visibility tools for OF@TEIN OpenFlow-based SDN testbed. In: 10th International Conference on Future Internet (CFI 2015), Seoul, Korea, June 2015
12. DevStack - an OpenStack Community Production, July 2015. http://docs.openstack.org/developer/devstack/
13. Medved, J., Varga, R., Tkacik, A., Gray, K.: OpenDaylight: towards a model-driven SDN controller architecture. In: Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, June 2014
14. Sherwood, R., et al.: Flowvisor: a network virtualization layer. Technical report Openflow-tr-2009-1, Stanford University, July 2009