

An OpenFlow-Based Energy-Efficient Data Center Approach

Michael Jarschel, Rastin Pries
University of Würzburg, Institute of Computer Science
Würzburg, Germany
{michael.jarschel|pries}@informatik.uni-wuerzburg.de

Categories and Subject Descriptors

C.2.1 [COMPUTER-COMMUNICATION NETWORKS]: Network Architecture and Design—*Network communications, Network topology*; C.4 [PERFORMANCE OF SYSTEMS]: Reliability, availability, and serviceability

Keywords

Data Center, OpenFlow, Energy-Efficiency

1. INTRODUCTION

Infrastructure as a service (IaaS) is one of the prevalent business models in cloud computing and has generated much customer interest over the past few years. An IaaS provider offers the temporary deployment and maintenance of a custom virtual host and network infrastructure to its customers on which arbitrary applications can be run and/or hosted. Providers of such a service face several challenges in their data centers. One of the main issues is the inherent heterogeneity of systems and applications from different customers. As a result, a variety of different load and traffic patterns has to be handled by the same data center infrastructure. There are two ways to remedy this. The first is the classical approach of overprovisioning to sustain a constant service quality. This is usually very inefficient and only open to companies with a huge budget and a lot of resources. The second one is smart resource management. An IaaS provider has to find a good balance between the various customer application requirements and the efficient use of the available resources in the data center. The ECDC (Energy efficient Data Center) approach we showcase in this demo is such a smart mechanism for finding this balance. It leverages monitoring information from machines as well as network devices and environmental data to create a coherent view of the current situation in a data center. This alone enables a single network operator to react to system changes as soon as they happen. However, the monitoring data is also used by a smart control application to react in certain situations by redistributing virtual machines, traffic flows and VLANs as well as powering devices up and down.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SIGCOMM'12, August 13–17, 2012, Helsinki, Finland.

Copyright 2012 ACM 978-1-4503-1419-0/12/08 ...\$15.00.

2. ARCHITECTURE

Figure 1 shows the ECDC architecture in a simple data center scenario. On the right we see the servers hosting the customers' virtual infrastructures. They are organized in racks with a top of the rack switch each. The entry point into the network from the service side are the virtual switches integrated into the hypervisor of each server. On the left we see two types of customers that are connecting to the data center network from the customer side, i.e. the Internet. The type A customer is a private home user who wants to use an entertainment service, e.g. video streaming, hosted by a service provider as a virtual infrastructure in rack B and C in the data center. The type B customer is a business user who uses a business application set up by his company in the data center, e.g. a virtual desktop infrastructure (VDI). As both types of customers have different demands and requirements, their traffic is kept in separate VLANs in the data center network to be able to manage them independently. The connectivity for type A customers is represented by red lines, for type B green lines are used. Packets are tagged at the entry switch into the data center network. The network itself is OpenFlow-enabled [3]. Forwarding decisions for all network elements, i.e. access, edge, and hypervisor switches are handled by a central entity - the OpenFlow controller (OFC). The control connection for each network element is established via a physically isolated management network. This network also connects the controller, OpenFlow switches, physical servers, and environmental sensors to the central data center management entity - the management station. The management station queries monitoring information on CPU-, network-, and memory-load, as well as power consumption from the connected devices via SNMP. Armed with this host of information, the management station generates the appropriate network policy for the OFC, distributes virtual ma-

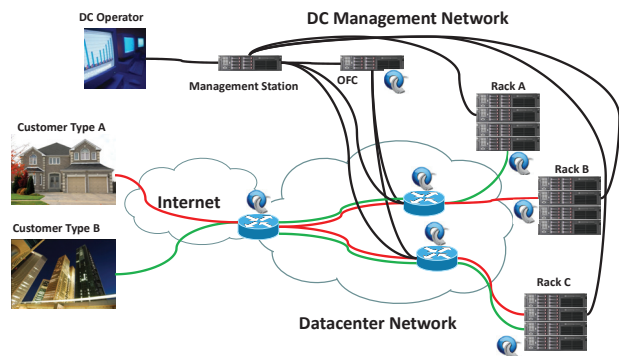


Figure 1: ECDC Architecture

chines across the servers, and powers down unused devices in order to ensure an efficient utilization of all resources while maintaining a good service quality for the customer at all times. The management station achieves this by observing a number of configured thresholds and timeouts for each service class. If a monitored parameter, e.g. CPU load, falls below or rises above a threshold for a certain amount of time as defined by a timeout, the management station will take action, e.g. by consolidating multiple virtual machines to one host or in the opposite case by spreading them over multiple hosts. Once such an action is triggered, the network, i.e. the OpenFlow controller, is immediately notified and can adapt the flow rules in the switches according to the new situation with little delay, minimizing the impact on the service. The gathered information is also presented to the data center operator through a graphical user interface as illustrated in Figure 2, which displays the current topology of the network and a time series of monitored values as configured by the operator. The operator is then able to facilitate changes in the operation of the data center, if this is necessary, e.g. through the introduction of a new service class.

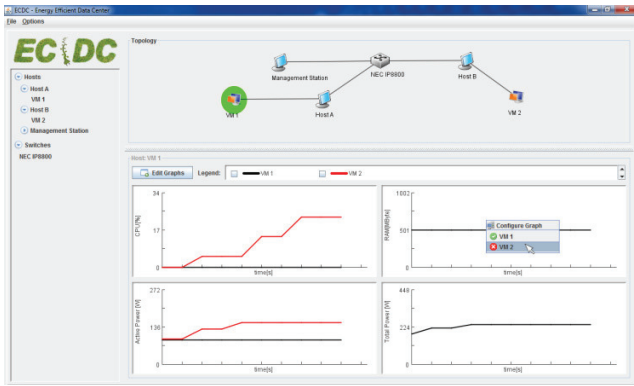


Figure 2: ECDC Operator GUI

3. DEMONSTRATION

The demonstration testbed is hosted on the German-Lab [6] (G-Lab) facility in Wuerzburg, Germany. Four rack servers are used as computing nodes running OpenNebula [1] and KVM [2] as hypervisor using the Open vSwitch [4] as virtual switch. The management station is hosted on a fifth server running the OpenNebula management software as well as our Java-based data center management software. The management network is a legacy IP network realized by the Cisco top of rack switch of the G-Lab facility. As OpenFlow controller, we use BigSwitch’s Floodlight hosted in a G-Lab virtual machine. The OpenFlow data center network is represented by a Pronto 3290 OpenFlow switch [5]. The demonstration will show the operation of the ECDC-enabled data center over the course of a business day. Using our own traffic generator, we emulate the behavior of the two types of users introduced in Section 2. For demonstration purposes we condensed the emulated “day” to a short cycle. During the progression of this cycle, we show the changes in the system as an operator would perceive

them using our ECDC GUI as our software adapts the resource allocation according to the demand. In the topology section of the GUI (cf. Figure 2) topology changes caused by the migration of virtual machines as well as the powering up and down of physical hosts will be displayed. In the monitoring section the collected information reflecting load changes in servers is illustrated by line graphs. If an OpenFlow switch is selected in the topology display, the monitoring section changes to show the switch’s flow table entries. By selecting an entry, the path of the corresponding flow through the network is displayed in the topology section.

4. CONCLUSION

In this demonstration we showcase our smart data center management software ECDC. It allows for an integrated adaption of computing and network resources according to the required capacity to ensure a smooth operation of services in an IaaS scenario. At the same time the software aims to minimize the carbon footprint of the data center in question by consolidating capacities and powering down those not needed. To achieve this, we leverage the modern software defined networking approach as well as proven open-source cloud management software. Challenges we are facing here are the integration of network and server virtualization by using the OpenVSwitch in combination with OpenNebula. We believe there is a lot of potential for optimization in data centers and this approach might also be applicable to a larger scale scenario.

5. ACKNOWLEDGMENTS

The authors would gratefully thank Thomas Höhn, Christopher Metter, and Phuoc Tran-Gia from the University of Würzburg for the work and fruitful discussions. This work was funded by the international bureau of the Federal Ministry of Education and Research (Förderkennzeichen VNM10/073).

6. REFERENCES

- [1] J. Fontán, T. Vázquez, L. Gonzalez, R. Montero, and I. Llorente. Opennebula: The open source virtual machine manager for cluster computing. In *Open Source Grid and Cluster Software Conference*, 2008.
- [2] A. Kivity, Y. Kamay, D. Laor, U. Lublin, and A. Liguori. kvm: the linux virtual machine monitor. In *Proceedings of the Linux Symposium*, volume 1, pages 225–230, 2007.
- [3] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. Openflow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2):69–74, 2008.
- [4] B. Pfaff, J. Pettit, T. Koponen, K. Amidon, M. Casado, and S. Shenker. Extending networking into the virtualization layer. *Proc. HotNets (October 2009)*, 2009.
- [5] Pica8. Pronto 3290 OpenFlow Switch. <http://www.pica8.org/products/p3290.php>.
- [6] D. Schwerdel, D. Günther, R. Henjes, B. Reuther, and P. Müller. German-lab experimental facility. *Future Internet-FIS 2010*, pages 1–10, 2010.