A new Framework to enable rapid innovation in Cloud Datacenter through a SDN approach.

José Teixeira

A thesis submitted to the University of Minho in the subject of Informatics, for the degree of Master of Science, under scientific supervision of Prof. Stefano Giordano and Prof. Alexandre Santos

University of Minho

School of Engineering

Department of Informatics

September, 2013

Acknowledgments

I would like...

I also...

Abstract

In the last years, the widespread of Cloud computing as the main paradigm to deliver a large plethora of virtualized services significantly increased the complexity of Datacenters management and raised new performance issues for the intra-Datacenter network. Providing heterogeneous services and satisfying userś experience is really challenging for Cloud service providers, since system (IT resources) and network administration functions are definitely separated.

In this scenario, a recent approach to programmable networks (i.e., Software-Defined Networking - SDN) seems to be a promising way to satisfy DC network requirements [1]. SDN based architecture decouples control and data planes: the most deployed SDN protocol is OpenFlow (OF) [2] [3], which allows to set into OF compliant switches forwarding rules established by a centralized intelligence called controller.

Since SDN allows to re-define and re-configure network functionalities (possibly up to the physical layer), the basic idea is to introduce a new framework that allows to develop and test new policies that enables a more efficient, agile, scalable and simple use of both server and network resources.

More specifically, the framework enhances both POX (an OF controller written in python) and Mininet (a well-known SDN emulator), with all the extensions necessary to experiment novel control and management strategies of IT and network resources.

... talk about obtained results and conclusions(not finished yet)

Keywords: Datacenter, Cloud, SDN, OpenFlow.

Contents

List of Acronyms

DC	Datacenter
SDN	Software Defined Networking
OF	Openflow
VM	Virtual Machine
IP	Internet Protocol
	Add as needed...

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

A Cloud Datacenter consists of virtualized resources that are dynamically allocated, in a seamless and automatic way, to a plethora of heterogeneous applications. In Cloud Datacenters, services are no more tightly bounded to physical servers, as occurred in traditional DCs, but are provided by Virtual Machines that can migrate from a physical server to another, increasing both scalability and reliability.

Software virtualization technologies allow a better usage of DC resources; DC management, however, becomes much more difficult, due to the strict separation between systems (*i.e.*, servers, VMs and virtual switches) and network (*i.e.*, physical switches) administration. Moreover, new issues arise, such as isolation and connectivity of VMs. Services performance may suffer from the fragmentation of resources as well as the rigidity and the constraints imposed by the intra-DC network architecture (usually a multilayer 2-tier or 3-tier fat-tree composed of Edge, Aggregation and Core switches [4]). Therefore, Cloud service providers (*e.g.*, [5]) ask for a next generation of intra-DC networks meeting the following features: 1) efficiency, *i.e.*, high server utilization; 2) agility, *i.e.*, fast network response to server/VMs provisioning; 3) scalability, *i.e.*, consolidation and migration of VMs based on applications' requirements; 4) simplicity, *i.e.*, performing all those tasks easily [6].

In this scenario, a recent approach to programmable networks (*i.e.*, Software-Defined Networking) seems to be a promising way to satisfy DC network requirements [1].

NOTE: To agressive this next sentense paragraph

SDN–based architecture decouples control and data planes: the most deployed SDN protocol is OpenFlow (OF) [3] [2], which allows to set into OF–compliant switches forwarding rules established by a centralized intelligence called controller. Since SDN allows to re-define and re-configure network functionalities (possibly up to the physical layer), the basic idea is to introduce an SDN cloud-DC controller that enables a more efficient, agile, scalable and simple use of both VMs and network resources. Nevertheless, before deploying the novel architectural solutions, huge test campaigns must be performed in experimental environments reproducing a real DC.

To this aim, we introduce a novel framework that enhances both Mininet [7] and POX [8] with all the software modules necessary to emulate an SDN-based intra-DC network, such as DC topology discovery, network traffic generation, etc. Specifically designed for DC environments, our framework allows to develop and assess novel SDN-Cloud-DC controllers, and to compare the performance of control and management strategies jointly considering both IT and network resources [9]. It is worth highlighting that the developed software modules may be ported in a real controller without changes, as our framework inherits such basic feature from Mininet.

The rest of the paper is organized as follows: section **??** provides a short survey of related works, whereas section **??** details the architecture and the functionalities of our framework. Section **??** presents an use case while section **??** evaluates the performance of the framework. Finally, section **??** concludes the paper with some final remarks.

- DataCenter
- Cloud
- Cloud DataCenter
- SDN - Software define Networks
- Openflow
- ...

1.2 Motivation and objectives

- Understanding the basic features of SDN paradigm
- Studying the problematics in cloud DC VM allocations

- Apply the SDN paradigm to better exploit the DC resources
- Develop a framework for Cloud Datacenter emulation and new VM allocation policies
- ...

1.3 Dissertation layout

In the present Chapter 1 - ...

Chapter 2

State of art

Usually background and related work ...

2.1 Available solutions

Write something generic

2.1.1 CloudSim

2.1.2 NetFPGA Emulation

2.1.3 Meridian

2.1.4 Networkcloudsim

2.1.5 Greencloud

2.1.6 icancloud

2.2 Virtual Machine Allocation Policies

Chapter 3

Architecture and design

Conceptual view and architecture of the proposed solution (implementation details can go into a different chapter, if required)...

3.1 Framework architecture

Generically talk about the architecture...

3.2 Framework modules: Mininet

3.2.1 Topology Generator

3.2.2 Traffic Generator

Describe each module, it's functionalities, limitations, how it can be used/improved (improved if the user wants to add new features)

- Talk generally about the traffic generator
- Talk about the one's we tried (pros and cons)

3.3 Framework modules: Controller

Describe each module, it's functionalities, limitations, how it can be used/improved (improved if the user wants to add new features)

3.3.1 Topology Discovery

3.3.2 OF Rules Handler

3.3.3 Statistics Handler

3.3.4 VM Request Handler

3.3.5 VMM - Virtual Machines Manager

3.3.6 Network Traffic Requester

3.3.7 POX Modules

3.3.8 User Defined Logic

3.4 Framework modules: Web Platform

Describe each module, it's functionalities, limitations, how it can be used/improved (improved if the user wants to add new features)

3.5 Framework modules: VM Requester

Describe each module, it's functionalities, limitations, how it can be used/improved (improved if the user wants to add new features)

3.6 Using the framework

3.6.1 Emulator

Describe how to use the framework (emulation part) and how to access the API..

3.6.2 Real Environment

Describe what changes in the real environment (the modules that are disabled and the ones that need to be enabled)

Chapter 4

Framework extensions

4.1 Enabling QoS

4.1.1 State of art: QoS in SDN

4.1.2 QoS in the framework

4.2 Enabling Virtual Machine migration

4.2.1 State of art: Virtual Machine Migration Policies

4.2.2 Virtual Machine migration in the framework

4.2.3 Usecase

Chapter 5

Validation and tests

Usually test and validation of the proposed solution ...

5.1 Framework Validation

- Show how Bf goes against WF with server driven algorithm (show server occupation)
- Show how Bf goes against WF with network driven algorithm (show network occupation) (although the behaviour is similar is allow to say that net algorithm may use switch statistics)

5.2 Performance Evaluation

Get the tests from the submitted paper.

5.3 Real environment tests

- Talk about the environment which was setup
  - Chosen hypervisor
  - Talk about Xen api and the alternative solution (ssh each server and run a script to clone the vm)
  - OpenVswitches VS NetFPGA problems
  -

Chapter 6

Conclusions

This chapter provides ...

6.1 Main contributions

6.2 Future work

Appendix A

Name of the Appendix

Bibliography

[1] J. Oltsik and B. Laliberte. Ibm and nec bring sdn/openflow to enterprise data center networks.

[2] Open Networking Foundation Home Page. *https://www.opennetworking.org.*

[3] OpenFlow Home Page. *http://www.openflow.org.*

[4] K. Bilal, S.U. Khan, J. Kolodziej, L. Zhang, K. Hayat, S.A. Madani, N. Min-Allah, L. Wang, and D. Chen. A comparative study of data center network architectures. In *European Conference on Modelling and Simulation*, 2012.

[5] AWS Home Page. *http://aws.amazon.com.*

[6] O. Baldonado, SDN, OpenFlow, and next-generation data center networks. *http://www.eetimes.com/design/embedded/43715 43/SDN–OpenFlow–and-next-generation-data-center-networks.*

[7] Mininet Home Page. *https://mininet.github.com.*

[8] POX Home Page. *http://ww.noxrepo.org.*

[9] D. Adami, B. Martini, G. Antichi, S. Giordano, M. Gharbaoui, and P. Castoldi. Effective resource control strategies using openflow in cloud data center. In *International Symposium on Integrated Network Management*. IEEE/IFIP, 2013.

[10] NS3. *http://www.nsnam.org.*

[11] Openstack Home Page. *http://www.openstack.org.*

[12] IBM Smart Cloud Provisioning Home Page. *http://www-01.ibm.com/software/tivoli/products /smartcloud-provisioning.*

[13] N. Calheiros, R. Ranjan, A. Beloglazov, C.A.F. De Rose, and R. Buyya. Cloudsim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software: Practice and Experience*, 41(1):23–50, 2011.

[14] K. Garg and R. Buyya. Networkcloudsim: Modelling parallel applications in cloud simulations. In *Internation Conference on Utility and Cloud Computing*. IEEE, 2011.

[15] J.D. Ellithorpe, Z. Tan, and R.H. Katz. Internet-in-a-box: Emulating datacenter network architectures using fpga's. In *Design Automation Conference*. ACM/IEEE, 2009.

[16] J. Matia, E. Jacob, D. Sanchez, and Y. Demchenko. An openflow based network virtualization framework for the cloud. In *International Conference on Coud Computing Technology and Science*. IEEE, 2011.

[17] M. Al-Fares, S. Radhakrishnan, B. Raghavan, N. Huang, and A. Vahdat. Hedera: Dynamic flow scheduling for data center networks. In *NSDI*. ACM, 2010.

[18] R. Raghavendra, J. Lobo, and K.W. Lee. Dynamic graph query primitives for sdn-based cloud network management. In *HotSDN*. ACM, 2012.

[19] A. Tavakoli, M. Casado, T. Koponen, and S. Shenker. Applying nox to the datacenter. In *HotNets*. ACM, 2009.

[20] A. Nunez, J.L. Vázquez-Poletti, A.C. Caminero, G.G. Castañé, J. Carretero, and I.M. Llorente. icancloud: A flexible and scalable cloud infrastructure simulator. *Journal of Grid Computing*, 2012.

[21] M. Banikazemi, D. Olshefski, A. Shaikh, J. Tracey, and G. Wang. Meridian: An sdn platform for cloud network services. *Communications Magazine*, 2013.

[22] S. Ostermann, K. Plankensteiner, R. Prodan, and T. Fahringer. Groudsim: An event-based simulation framework for computational grids and clouds. *Euro-Par 2010 Parallel Processing Workshops*, 2011.

[23] D. Kliazovich, P. Bouvry, and S.U. Khan. Greencloud: A packet-level simulator of energy-aware cloud computing data centers. In *Globecom*. IEEE, 2010.

[24] A. Pescapè A. Dainotti, A. Botta. A tool for the generation of realistic network workload for emerging networking scenarios. *Computer Networks (Elsevier), 2012, Volume 56, Issue 15, pp 3531-3547*, 2012.