PhD in Information Technology and Electrical Engineering

Università degli Studi di Napoli Federico II

PhD Student: Luigi De Simone

XXIX Cycle

Training and Research Activities Report – Third Year

Tutor: Prof. Domenico Cotroneo

1. Information

PhD candidate: Luigi De Simone
Date of birth: 24/02/1986
Master Science title: Master's degree in Computer Engineering (cum laude), Universiy of Naples Federico II
Doctoral Cycle: XXIX
Fellowship type: PhD student grant
Tutor: Prof. Domenico Cotroneo
Year: Third

I received my MS degree (cum laude) in Computer Engineering from the Univesità degli Studi di Napoli Federico II in July 2013.

My master thesis focused on the dependability of the Linux OS, specifically the fault-tolerance of device drivers. Device drivers are software components with the most of the defects ("bugs") within an operating system, thus they are the main cause of operating system failures. I proposed a novel fault-tolerance approach based on run-time monitoring and fault-detection of a storage device driver, and I developed and tested the approach to a storage device driver of the Linux kernel.

I'm currently at third year of PhD program in Information Technology and Electrical Engineering (ITEE) at Federico II University of Naples, under the supervision of Prof. Domenico Cotroneo.

2. Study and Training activities

In this third year I attended the following courses and seminars.

Title	Type	Hours	Credits	Dates	Organizer	Certificate
Communicating and disseminating your research work	Ad-Hoc Module	18 hours (3 days)	3	14-15-16/03/2016	Università degli Studi di Napoli Federico II	Yes
Modelling Critical Infrastructures (CIs) Resilience	Seminary	2h	0.40	03/112016	Università degli Studi di Napoli Federico II	Yes
Security Operations in una Telco, esperienze e riflessioni dal campo	Seminary	2h	0.40	11/11/2016	Università degli Studi di Napoli Federico II	Yes
MINIX3: A Reliable and Secure Operating System	Seminary	1.5h	0.20	30/11/2016	Università degli Studi di Napoli Federico II	Yes

Student: Name Surname **Cycle XXIX**

luigi.desimone@unina.it

	Credits year 1		Credits year 2		Credits year 3								
						1	2	3	4	5	6		
	Estimated	Summary	Estimated	Summary	Estimated	bimonth	bimonth	bimonth	bimonth	bimonth	bimonth	Summary	Total	Check
Modules	20	23	10	6	9	3	0	0	0	0	0	3	32	30-70
Seminars	5	7,3	5	1,8	0,9	0	0	0	0	1	0	1	10,1	10-30
Research	35	37	45	52,2	50,1	7	9	8	8	9	9	50	139,2	80-140
	60	67,3	60	60	60	10	9	8	8	10	9	54	181,3	180

3. Research activity

Title: Dependability Benchmarking in Network Function Virtualization

In this first year of my PhD, the first goal of my research it has been to study and to understand which are the challenges and open problems behind the evaluation of a cloud computing ecosystems (CCE) **dependability**. As I have depicted in my study presented at *IEEE International Symposium on Software Reliability Engineering Workshops* (ISSREW, Napoli 2014) **[P1]**, it is necessary to conduct research and develop techniques and methodologies that allow us to build countermeasures against faults, with the purpose of preventing fault propagation within a CCE and, ultimately, of avoiding failures of the CCE as a whole.

During the second and third second year I have collaborated with a global leader company of TLC solutions, within an industrial research project with the objective to propose methodology and tools to evaluate dependability of Network Function Virtualization systems.

Network Function Virtualization (NFV) [1], [2] is an emerging solution to supersede traditional network equipment to reduce costs, improve manageability, reduce time-to-market, and provide more advanced services [3]. NFV will exploit IT virtualization technologies to turn network equipment into *Virtualized Network Functions* (VNFs) that will be implemented in software, and will run on commodity hardware, virtualization and cloud computing technologies located in high-performance data centers, namely *Network Function Virtualization Infrastructures* (NFVIs). Thus, NFVI can be seen as a complex cloud computing infrastructures.

NFV solutions have to compete not only in cost and manageability, but also in **performance** and **reliability**: telecom regulations impose **carrier-grade** requirements to network functions, which need to achieve extremely low packet processing overheads, controlled latency, and efficient virtual switching, along with quick and automatic recovery from faults (in the order of few seconds) and extremely high availability (99.999% or higher) [4]. It is well known that these requirements are well satisfied by traditional (hardware-based) network functions, which have been proven very reliable over the last decades. However, performance and reliability is definitively a big challenge to achieve in next generation of network functions, where most of the control logic will be implemented by means of software and virtualization technologies.

The need for Dependability Benchmarking Methodology in NFV

Benchmarks are an established practice for performance evaluation in the computer industry since decades. Examples of successful benchmarking initiatives are the TPC (Transaction Processing Performance Council [5]) and the SPEC (Standard Performance Evaluation Corporation [6]).

More recently, the research community developed dependability benchmarking procedures, which have significantly matured from both the methodological and from technical point of view [7, 8, 9, 10]. However, dependability benchmarking is a more difficult task than performance and functional benchmarking, as it needs to consider the presence of faults in the system, which requires elaborated test scenarios and experimental procedures, by leveraging on dependability evaluation techniques (in particular, fault injection).

Dependability benchmarking becomes more compelling for NFV, as denoted by the interest of standardization bodies to define reliability requirements and evaluation procedures for the cloud and for NFV [11, 12], and also the effort to drive the consistent implementation of an open and standard NFV reference platform [13]. As mentioned, the need for dependability benchmarking is exacerbated by the high incidence

of the faults, due to the large scale and complexity of NFVIs, the dynamism of NFV services, and the massive adoption of commercial off-the-shelf (COTS) hardware and software components. While COTS components are easily procured and replaceable, NFV will need to recover from faulty components in a timely way and preserve high network performance. However, there is a lack of approaches that could allow NFV vendors, providers and users to evaluate dependability, with a degree of accuracy and trustworthiness comparable to performance benchmarks.

During the second year, I presented the research paper **[P4]** we show a preliminary dependability evaluation and benchmarking methodology for NFV. Based on fault injection, the methodology analyzes how faults impact on VNFs in terms of performance degradation and service unavailability, and in terms of effectiveness of the implemented fault management mechanisms.

In this third year, I've analyzed more deeply alternative virtualization technologies to adopt NFV and define the dependability benchmarking methodology.

We provide context, measures, and faultloads to conduct dependability benchmarks in NFV, according to the general principles of dependability benchmarking, i.e., representative, simple, and portable across alternative technologies [14]. Moreover, the benchmark takes into account the NFV use cases and technologies. The benchmark allows to: (i) get quantitative measures of worst-case quality of service; (ii) identify which fault types and faulty components impact most on NFV services; (iii) validate the effectiveness of fault tolerance and high-availability algorithms and mechanisms.

The faultload is obtained by modeling the virtualized infrastructure for the four domains according to a typical virtualized infrastructure: network, storage, CPU, and memory. These elements are present both as virtual resource abstractions, and as physical resources. The fault model is aimed to the benchmark performer, which builds the faultload for the target infrastructure by systematically applying the fault model to each resource in the infrastructure (virtual and physical machines, virtual and physical disks and switches). For each domain (physical and virtual CPU, memory, disk and network), we identify faults according to the following three general fault types: unavailability (the resource becomes unresponsive and unusable); delay (the resource is overcommitted and slowdown); corruption (the information stored or processed by the resource is invalid). These fault types are broad classes that span over the possible failure modes of a component [15, 16, 17, 18, 19]. We specialize these general fault types for each resource, by analyzing how hardware, software, and/or operator faults can likely cause these three possible fault types [14]. In this analysis, we consider the scientific literature on fault injection and failure analysis in cloud computing infrastructures [20, 21, 22, 23, 24, 25, 26], well-known cloud computing incidents [27, 28, 29], and knowledge on the prospective architecture and products for NFVI [30], to identify a representative and complete set of faults.

Furthermore, during the third year I implemented a **Fault Injection toolsuite** for virtualization technologies assessment. Dependability benchmarking requires fault injection testing technologies to support the experimental evaluation. Therefore, during this year I have designed and implemented fault injection tools for both for **VMware vSphere** and **Docker**, by adopting the fault model defined by the dependability benchmarking methodology. The faults are injected by emulating their effects on the virtualization layer. In particular, I/O and compute faults can be emulated, respectively, by deliberately injecting I/O losses, corruptions and delays, and by injecting code and data corruptions in memory and in CPU registers, by forcing a crash of VMs and of their hosting nodes, and by introducing CPU- and memory-bound "hogs" (that is, tasks that deliberately consume CPU cycles and allocate memory areas) in order to cause resource exhaustion. These faults can be injected either in a specific virtual domain (a VM or a container), or in a physical machine (PM) of the NFVI.

Finally, I have analyzed an NFV **case study** on dependability benchmarking. In the experimental case study I have compared two candidate virtualization technologies for NFV: the commercial, hypervisor-based virtualization platform **VMware vSphere**, and the open-source, container-based virtualization platform **Docker**. We evaluate these technologies in the context of an high-availability, NFV-oriented IP Multimedia Subsystem (IMS), which has been deployed on two alternative NFVI configurations.

4. Products

During the three years, I have produced the following products.

Conference Paper

[P1] **De Simone, L.**, "Towards Fault Propagation Analysis in Cloud Computing Ecosystems," Software Reliability Engineering Workshops (ISSREW), 2014 IEEE International Symposium on , pp.156,161, 3-6 Nov. 2014, DOI: 10.1109/ISSREW.2014.47

BEST PRESENTATION AWARD

[P2] Cotroneo, D.; **De Simone, L.**; Iannillo, A.K.; Lanzaro, A.; Natella, R.; Jiang Fan; Wang Ping, "Network Function Virtualization: Challenges and Directions for Reliability Assurance," Software Reliability Engineering Workshops (ISSREW), 2014 IEEE International Symposium on , pp.37,42, 3-6 Nov. 2014, DOI: 10.1109/ISSREW.2014.48

[P3] Cotroneo, D.; **De Simone, L.**; Iannillo, A.K.; Lanzaro, A.; Natella, R., "Improving Usability of Fault Injection," Software Reliability Engineering Workshops (ISSREW), 2014 IEEE International Symposium on, pp.530,532, 3-6 Nov. 2014, DOI: 10.1109/ISSREW.2014.37

[P4] Domenico Cotroneo, **Luigi De Simone**, Antonio Ken Iannillo, Anna Lanzaro, Roberto Natella, "*Dependability Evaluation and Benchmarking of Network Function Virtualization Infrastructures*", at 1st IEEE Conference on Network Softwarization (NetSoft), pp. 1 – 9, 13-17 April 2015 London, DOI: 10.1109/NETSOFT.2015.7116123
BEST PAPER AWARD

[P5] Domenico Cotroneo, **Luigi De Simone**, Francesco Fucci, Roberto Natella, "*MoIO: Run-time monitoring for I/O protocol violations in storage device drivers*", at Software Reliability Engineering (ISSRE), 2015 IEEE 26th International Symposium on, pp. 472 – 483, 2-5 Nov. 2015 Gaithersbury, MD, DOI: 10.1109/ISSRE.2015.7381840

Journal Paper

[P6] Domenico Cotroneo, **Luigi De Simone**, Roberto Natella, "*NFV-Bench: A Dependability Benchmark for Network Function Virtualization Services*", IEEE Transaction on TBD, *IN PREPARATION*

5. Conferences

None

6. Tutorship

During the three years I have been teaching assistant for the course of Operating Systems. Furthermore, I was giving some lectures in the context of course of Distributed Systems, a.a. 2015/2016. Furthermore, I have been MSc thesis co-advisor on topic about verification of fault tolerant mechanisms in NFV.

References

[1] NFV ISG, "Network Functions Virtualisation - An Introduction, Benefits, Enablers, Challenges & Call for Action," ETSI, Tech. Rep., 2012.
[2] NFV ISG, "Network Functions Virtualisation (NFV) - Network Operator Perspectives on Industry Progress," ETSI, Tech. Rep., 2013.
[3] A. Manzalini, R. Minerva, E. Kaempfer, F. Callegari, A. Campi, W. Cerroni, N. Crespi, E. Dekel, Y. Tock, W. Tavernier *et al.*, "Manifesto of edge ICT fabric," in *Proc. ICIN*, 2013, pp. 9–15.
[4] Quality Excellence for Suppliers of Telecommunications Forum (QuEST Forum). TL 9000 Quality Management System Measurements Handbook 4.5. Technical report, 2010.
[5] Transaction Processing Performance Council. TPC Homepage. http:// www.tpc.org/.
[6] Standard Performance Evaluation Corporation. SPEC Homepage. https: //www.spec.org/.
[7] J. Durães, M. Vieira, and H. Madeira. Multidimensional Characterization of the Impact of Faulty Drivers on the Operating Systems Behavior. IEICE Trans. on Inf. and Sys., 86(12):2563–2570, 2003.
[8] P. Koopman and J. DeVale. The Exception Handling Effectiveness of POSIX Operating Systems. IEEE Trans. on Software Engineering, 26(9):837–848, 2000.
[9] J. Durães and H. Madeira. Generic Faultloads based on Software Faults for Dependability Benchmarking. In Proc. IEEE/IFIP Intl. Conf. Dependable Systems and Networks, pages 285–294, 2004.
[10] M. Vieira and H. Madeira. A dependability benchmark for OLTP application environments. In Proc. Intl. Conf. on Very Large Data Bases, pages 742–753, 2003.
[11] European Union Agency for Network and Information Security. Cloud computing certification.
[12] Network Functions Virtualisation (NFV) - Network Operator Perspectives on Industry Progress. White Paper, 2013.
[13] Christofer Price and Sandra Rivera. Opnfv: An open platform to accelerate nfv. White Paper. A Linux Foundation Collaborative Project, 2012.
[14] DBench project. DBench Final Report. http://www.laas.fr/DBench/, 2004.
[15] D. Powell. Failure Mode Assumptions and Assumption Coverage. In Proc. Intl. Symp. on Fault-Tolerant Comp., pages 386–395, 1992.
[16] A. Mukherjee and D.P. Siewiorek. Measuring software dependability by robustness benchmarking. IEEE Trans. on Software Engineering, 23(6):366– 378, 1997.
[17] J.H. Barton, E.W. Czeck, Z.Z. Segall, and D.P. Siewiorek. Fault Injection Experiments using FIAT. IEEE Trans. on Computers, 39(4):575–582, 1990
[18] F. Cristian. Exception handling and software fault tolerance. IEEE Trans. on Computers, C-31(6):531–540, 1982.
[19] JJ Hudak, B.H. Suh, DP Siewiorek, and Z. Segall. Evaluation and Comparison of Fault-Tolerant Software Techniques. IEEE Trans. on Reliability, 42(2):190–204, 1993.
[20] Xiaoen Ju, Livio Soares, Kang G. Shin, Kyung Dong Ryu, and Dilma Da Silva. On fault resilience of OpenStack. In Proceedings of the 4th annual Symposium on Cloud Computing - SOCC '13, pages 1–16, New York, New York, USA, October 2013. ACM Press.
[21]Haryadi S. Gunawi, Thanh Do, Pallavi Joshi, Peter Alvaro, Joseph M. Hellerstein, Andrea C. Arpaci-Dusseau, Remzi H. Arpaci-Dusseau, Koushik Sen, and Dhruba Borthakur. FATE and DESTINI: a framework for cloud recovery testing. page 18, March 2011.
[22] Pallavi Joshi, Haryadi S. Gunawi, and Koushik Sen. PREFAIL. In Proceedings of the 2011 ACM international conference on Object oriented programming systems languages and applications - OOPSLA '11, volume 46, page 171, New York, New York, USA, October 2011. ACM Press.
[23] Cuong Pham, Daniel Chen, Zbigniew Kalbarczyk, and Ravishankar K.Iyer. CloudVal: A framework for validation of virtualization environment in cloud infrastructure. In 2011 IEEE/IFIP 41st International Conference on Dependable Systems & Networks (DSN), pages 189–196. IEEE, June 2011.
[24] Michael Le and Yuval Tamir. Fault injection in virtualized systems—challenges and applications. Dependable and Secure Computing, IEEE Transactions on, 12(3):284–297, 2015.
[25] Michael Le and Yuval Tamir. Rehype: Enabling vm survival across hypervisor failures. In ACM SIGPLAN Notices, volume 46, pages 63–74, 2011.

[26] Nadav Amit,Dan Tsafrir,Assaf Schuster,Ahmad Ayoub,and Eran Shlomo. Virtual cpu validation. In Proceedings of the 25th Symposium on Operating Systems Principles, pages 311–327. ACM, 2015.
[11] Amazon.com, Inc. Summary of the amazon ec2 and amazon rds service
disruption in the us east region, April 2011.
[60] Gigaom.com. Windows azure outage hits europe, July 2012.
[156] Alan Warren. What happened to google docs on wednesday, September 2011.
[151] Inc. VMware. VMware vSphere 6 Fault Tolerance - Architecture and Performance. Technical report, 2016.