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 – First Year

Tutor: 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: First

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 first 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 first year I attended the following courses and seminars.

Title	Type	Hours	Credits	Dates	Organizer	Certificate
Sistemi Real-time	Master Science Course (ING-INF/05)		6	2nd semester 2013-2014	Università degli Studi di Napoli Federico II	Yes
Impianti di Elaborazione	Master Science Course (ING-INF/05)		9	1st semester 2014-2015	Università degli Studi di Napoli Federico II	Yes
The Entrepreneurial Analysis of Engineering Research Projects	Ad-hoc		3	9, 11, 18, and 20 February 2015	Università degli Studi di Napoli Federico II	Yes
EuroProgettazione	Ad-hoc		3	17, 20, 24, 30 Oct. 2014	Università degli Studi di Napoli Federico II	Yes
Project Management	Ad-hoc		3	30 Jan. 2015, 6,13,20 and 27 Feb. 2015	Università degli Studi di Napoli Federico II	To be completed
11th International School on Software Engineering	Doctoral School		3	June 30th to July 3rd	Univesità degli Studi di Salerno	Yes
Reliability and Availability Modeling in Practice, by Prof. Kishor Trivedi	Seminary	2	0,4	05/11/2014	DIETI	Yes
Capacity Planning for Infrastructure-as-a-Service Cloud, by Prof. Kishor Trivedi	Seminary	2	0,4	07/11/2014	DIETI	Yes
Methods and tools for smart device integration and simulation, by Prof. Franco	Seminary	2	0,4	20/11/2014	DIETI	Yes

Fummi						
UML Profiles for the specification of non functional properties of software systems, by Prof.ssa Simona Berardi	Seminary	2,5	0,5	26/11/2014	DIETI	Yes
Site Reliability Engineering at Google", by Ph.D. Marco Papa Manzillo	Seminary	3	0,6	27/11/2014	DIETI	Yes
Seminar on memory technologies for Android based systems, by Dott. Luca Porzio and Dott. Graziano Mirichigni	Seminary	2	0,4	05/12/2014	DIETI	Yes
Applications for software development: types, interactions and continuous integration, by Dott. Antonio Almazàn	Seminary	2	0,4	16/01/2015	DIETI	Yes
Risk management meets model checking: fault tree analysis and model-based testing via games, by Prof.ssa Mariëlle Stoelinga	Seminary	2	0,4	20/01/2015	DIETI	Yes
Joint location and design optimization for resource allocation in software-defined virtual networks, by Proff. Antonia Tulino e Claudio Sterle	Seminary	2	0,4	21/01/2015	DIETI	Yes
Linked Open Data-enabled Strategies for Top-N Recommendations, by Dott. Pierpaolo Basile	Seminary	1,5	0,3	05/02/2015	DIETI	Yes
Three core issues for the Internet: things, security and economics, by Prof. Henning Schulzrinne	Occasionally provided module	8	2	19-20 Feb. 2015	DIETI	Yes

Student: Name Surname luigi.desimone@unina.it

Tutor: Name Surname cotroneo@unina.it

	Credits year 1								Credits year 2	Credits year 3		
		1	2	3	4	5	6					
	Estimated	bimonth	bimonth	bimonth	bimonth	bimonth	bimonth	Summary	Estimated	Estimated	Total	Check
Modules	20			6		3	14	23	10	0	33	30-70
Seminars	5			3		2,7	1,6	7,3	5	0	12,3	10-30
Research	35	8	10	2	9	6	2	37	45	60	142	80-140
	60	8	10	11	9	11,7	17,6	67,3	60	60	187,3	180

3. Research activity

Title: Dependability issues in cloud computing infrastructures

Description and Study

3.1 What is Cloud Computing?

Cloud Computing is strongly reshaping the IT industry panorama, and imposes many research challenges for the near future. Such impact was shown in a recent IDC market analysis [1], which predicted that the cloud software market will surpass $75B by 2017, attaining a five year compound annual growth rate of 22% in the forecast period.

Cloud computing was born to avoid the need for many corporations to build and manage their own IT data centers: it is a new paradigm meant to provide computational resources, in such a way to allocate and deallocate them on demand, based on a pay-per-use business model, avoiding expensive hardware platforms and big initial capital costs. Such computing resources include Internet services, storage facilities, and computing power, all provided just like a service. In this panorama, several research projects have been pursuing novel methodologies, architectures and tools to achieve trustworthy cloud services [2] [3].

3.2 Dependability in cloud computing

A cloud computing infrastructure is a complex *ecosystem* (Cloud Computing Ecosystems – CCE), in which several actors play a crucial role. These ecosystems are highly distributed, consist of heterogeneous hardware and software components, and are expected to provide highly-available services requested by millions of user in parallel. Failures in such a complex system are inevitable, because of too many factors are outside of our control. They include not only hardware faults (e.g., CPU, memory, disk, network faults) but also software faults (i.e., bugs) and operator faults (e.g., a software misconfiguration). In these infrastructures mostly are driven by software, thus the likelihood of incurring in software and operator faults is very high, leading cloud services to outages, unresponsiveness and data losses.

Delivering a *trustworthy* cloud computing service (ranging from IaaS to SaaS [8]) is a priority. Indeed, if we think that a lot of organizations and companies rely on cloud computing services (e.g., in traffic management system for decision support [6], in finance [7], in healthcare [8], in network virtualization [4, 5]), cloud services will become more critical in the near future!

Failures are random in time and it is really difficult to predict them. In addition, many cloud services depend on third-party services over the Internet, exposing cloud services to cascading failures. Given such an unpredictable and dynamic scenario, it is extremely important to know the nature of faults and their propagation within CCE, in order to deliver dependable cloud services. Addressing dependability issues early in design and making decisions to reduce the impact of failures of a specific service, are important benefits for cloud developers and designers.

3.3 Dependability evaluation of cloud computing infrastructure

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 infrastructure **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.

Furthermore, it is important to provide techniques and methodologies for understanding how faulty components in the CCE can affect other components and the overall CCE services, and for predicting and quantifying the impact of fault propagation on the CCE as a whole.

Recent studies have been done in testing of cloud-based applications [9], using cloud platforms to perform testing of application [10], and studies related to verification of cloud services [11]. Furthermore, other studies [2] addressed the problem of reliability of cloud infrastructure.

Nevertheless, there is still a need for approaches specifically focused on the reliability of cloud services and infrastructures against faults.

In recent years, several studies and tools have proved that **Fault Injection Testing** is a valuable approach for assessing fault-tolerant systems [12]. Fault injection is an approach in which we deliberately introduce faults in a system. This approach can assess the robustness and performance of a system in the presence of faults, and to state if fault tolerance algorithms and mechanisms are effective.

In the CCE context, **Virtualization** is cornerstone technology to set up a CCE. Virtualization allows to abstract physical resources (e.g., CPUs, network devices, storage devices, etc.) in order to share and to provide resources, making a physical machine as a soft component to use and manage very easily.

Thus, to assure the dependability of cloud systems, it is necessary to assess the reliability of the virtualization environment as a whole, focusing both on VMs and on the Hypervisor, as well as on the Cloud Management Stack software that orchestrates them (such as the well-known OpenStack framework) to efficiently manage cloud infrastructures.

Recent studies have faced with testing of components that constitute CCE, adopting fault injection to assure a high-level of reliability of cloud systems. These studies mainly focusing on Virtual Machines (*D-Cloud* [13]; *DS-Bench Toolset* [14], *Chaos Monkey* [15]), Hypervisors (*CloudVal* [16]), and cloud management stack (*Openstack resilience study* [17], *PreFail* [18]). All these studies are mostly focused on injection of hardware faults, for example CPU (e.g., corrupt registers), memory (e.g., bit error), network controller (e.g., bit error in packet), and hard disk (e.g., fault in a specific sector). Moreover, these tools are not meant for the evaluation of CCE architecture as a whole.

In addition to hardware faults, a system can be affected also by software faults and configuration faults. A fundamental part of fault injection testing is the definition of the *Fault Model*, which is a description of the types of fault that the system is expected to experience during runtime. It drives fault injection tests specifying *what* to inject, *when* to inject and *where* to inject. It is very challenging to define realistic fault models that take into account all the specifics of each CCE elements, given the complexity of these systems. Furthermore, in the CCE context, software and operator faults have not yet been studied deeply, thus there is another big question to answer.

Failures in CCEs may involve fault propagation, and due to complex interactions between different layers, it is very challenging to predict and quantify which is the impact that such a propagation could have on the CCE as a whole.

The idea is to leverage fault injection techniques to conduct such a fault propagation analysis. We can inject faults (hardware, software and configuration faults) in each layer, to understand how these faults propagate through different components and layers within CCEs. This analysis can give useful information about if there is (or not) a fault propagation path from less critical components/layers to more critical components/layers. Furthermore, we can discover new failure modes, and localize failures to the greatest extent possible. This work aims to develop framework, tools, mechanisms, and algorithms in order to detect faults and prevent their propagation within CCEs.

3.4 Dependability evaluation of NFV infrastructures

During my first year I also collaborate with Huawei Technologies Co. Ltd., 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) [4], [5] is an emerging solution to supersede traditional network equipment to reduce costs, improve manageability, reduce time-to-market, and provide more advanced services [19]. 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).

In particular, within my research group, I have studied how to assess the risks introduced by virtualization technologies for NFVI reliability [P3]. Towards this goal, we conduct the following activities:

- **Failure Mode and Effects Analysis of virtualization technologies in NFVIs**: we need to analyze the architecture of NFVI and its potential threats in order to understand what can affect reliability. The FMEA should consider not only hardware failures, but also failures due to software and configuration faults that can impact on virtualized resources (*e.g.*, virtual CPU, memory, network and storage);
- **Definition of Key Performance Indicators and Methodologies for NFVI reliability**: we will define measures for fault tolerance and performance, and provide guidelines to allow reliability engineers to systematically assess reliability by means of fault injection testing;
- **Design of novel Fault Injection Techniques**: because of the challenges in NFVIs (e.g., black-box technologies), the most advantageous injection target seems to be represented by the interfaces of the Compute, Hypervisor and Network domains. The errors and corruptions to be injected should be defined on the basis of the FMEA;
- **Validation using NFV products and technologies**: we will conduct a proof-of-concept validation of the fault injection approach on commercial NFV products, based on virtualization technologies such as VMware and LXC.

In a study under review [P4], we show a dependability evaluation and benchmarking methodology for NFVIs. Based on fault injection, the methodology analyzes how faults impact on VNFs in terms of performance degradation and service unavailability. The case study on the IMS showed how the methodology can point out dependability bottlenecks in the NFVI and guide design efforts.

4. Products

In this first year, I have produced the following products.

4.1 Publications

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", **submitted** at 1st IEEE CONFERENCE ON NETWORK SOFTWARIZATION (NetSoft)

5. Conferences

I participated the following conference:

Conference name	Place	Dates	Number of papers
The 25th IEEE International Symposium on Software Reliability Engineering (ISSRE)	Naples	November 3-6, 2014	113

As the author, I **presented** the following paper:

- *Towards Fault Propagation Analysis in Cloud Computing Ecosystems*, at 2014 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), Naples (see [P1])

6. Tutorship

In this first year I have been teaching assistant for the course of Operating Systems, a.a. 2014/2015

References

[1] IDC and Cisco. Midsize enterprises leading the way with cloud adoption. [Online]. Available: http://share.cisco.com/cloudadoption/

[2] A. Bessani, R. Kapitza, D. Petcu, P. Romano, S. V. Gogouvitis, D. Kyriazis, and R. G. Cascella, “A look to the old-world sky: EU- funded dependability cloud computing research,” *SIGOPS Operating Systems Review*, vol. 46, no. 2, pp. 43–56, Jul. 2012.

[3] Microsoft Corporation. Trustworthy computing homepage. [Online]. Available: http://www.microsoft.com/en-us/twc/default.aspx

[4] NFV ISG, “Network Functions Virtualisation - An Introduction, Benefits, Enablers, Challenges & Call for Action,” ETSI, Tech. Rep., 2012.

[5] ——, “Network Functions Virtualisation (NFV) - Network Operator Perspectives on Industry Progress,” ETSI, Tech. Rep., 2013.

[6] Z. Li, C. Chen, and K. Wang, “Cloud computing for agent-based urban transportation systems,” *Intelligent Systems, IEEE*, vol. 26, no. 1, pp. 73–79, Jan 2011.

[7] “Cloud computing for financial markets,” White Paper, Cisco Systems, Inc., 2011. [Online]. Available: http://www.cisco.com/web/strategy/ docs/finance/cloud wp c112D518876.pdf

[8] “Your cloud in healthcare,” White Paper, VMware, Inc., 2011. [Online]. Available: http://www.vmware.com/files/pdf/ VMware- Your- Cloud- in- Healthcare- Industry- Brief.pdf

[9] X. Bai, M. Li, B. Chen, W.-T. Tsai, and J. Gao, “Cloud testing tools,” in *Proc. Intl. Symp. SOSE*, 2011, pp. 1–12.

[10] L. Ciortea, C. Zamfir, S. Bucur, V. Chipounov, and G. Candea, “Cloud9: A software testing service,” *SIGOPS Operating System Review*, vol. 43, no. 4, pp. 5–10, Jan. 2010.

[11] S. Bouchenak, G. Chockler, H. Chockler, G. Gheorghe, N. Santos, and A. Shraer, “Verifying cloud services: Present and future,” *SIGOPS Operating System Review*, vol. 47, no. 2, pp. 6–19, Jul. 2013.

[12] R. Natella, D. Cotroneo, J. Duraes, and H. Madeira, “On fault repre- sentativeness of software fault injection,” *Software Engineering, IEEE Transactions on*, vol. 39, no. 1, pp. 80–96, Jan 2013.

[13] T. Banzai, H. Koizumi, R. Kanbayashi, T. Imada, T. Hanawa, and M. Sato, “D-cloud: Design of a software testing environment for reliable distributed systems using cloud computing technology,” in *Proc. Intl. Conf. CCGRID*, 2010, pp. 631–636.

[14] H. Fujita, Y. Matsuno, T. Hanawa, M. Sato, S. Kato, and Y. Ishikawa, “DS-Bench Toolset: Tools for dependability benchmarking with simu- lation and assurance,” in *Proc. Intl. Conf. DSN*, 2012, pp. 1–8.

[15] Netflix. The Chaos Monkey. [Online]. Available: https://github.com/ Netflix/SimianArmy/wiki/Chaos- Monkey

[16] C. Pham, D. Chen, Z. Kalbarczyk, and R. K. Iyer, “CloudVal: A framework for validation of virtualization environment in cloud infras- tructure,” in *Proc. Intl. Conf. DSN*, 2011, pp. 189–196.

[17] X. Ju, L. Soares, K. G. Shin, K. D. Ryu, and D. Da Silva, “On fault resilience of OpenStack,” in *Proc. SOCC*, 2013, pp. 1–16.

[18] P. Joshi, H. S. Gunawi, and K. Sen, “Prefail: A programmable tool for multiple-failure injection,” in *Proc. Intl. Conf. OOPSLA*, 2011, pp. 171–188.

[19] 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.