PhD in Information Technology and Electrical Engineering

Università degli Studi di Napoli Federico II

PhD Student: Innocenzo Mungiello

XXXI Cycle

Training and Research Activities Report – Second Year

Tutor: Alessandro Cilardo

1. Information

a. Innocenzo Mungiello, master's degree in Computer Engineering, magna cum laude, in 2015 from the University of Naples Federico II.
b. XXXI Cycle – ITEE
c. No Grant
d. Tutor: Alessandro Cilardo

2. Study and Training activities

a. MS Modules:
    i. *Circuiti per DSP*, Davide De Caro, 16/12/2016, 9 CFU;
    ii. *Sistemi Embedded*, Antonino Mazzeo, 22/09/2017, 6 CFU;
b. Ad Hoc Modules:
    i. *Testing Automation*, Porfirio Tramontana, 02/02/2017, 3 CFU;
c. Seminars
    i. *Modelling Critical Infrastructures (CIs)*, Peter Popov, 3/11/2016, 0.4 CFU;
    ii. *Diritti Umani e Nuove Tecnlogie,* Daniele Amoroso, 9/11/2016, 0.5 CFU;
    iii. *Software Design diversity from conceptual models to practical implementations,* Peter Popov, 21/11/2016, 0.4 CFU;
    iv. *Minix 3: A reliable and secure Operating System,* A. Tanembaum, 30/11/2016, 0.4 CFU;
    v. *Video Coding / Transcoding in Heterogeneous HPC,* Mario Kovac, 7/12/2016, 0.4 CFU;
    vi. *Plasma stability and dynamic events in Tokamaks with a resistive wall,* Dr. Vladimir Dmitrievich Pustovitov, 30/01/2017, 0.4 CFU;
    vii. *IBM Cognitive Computing: challenges and opportunities in building an Artificial Intelligence platform for business,* Pietro Leo, 17/02/2017, 0.4 CFU;
    viii. *Cognitive Computing and da Vinci robot: Research proposal and discussion,* Paolo Maresca, 17/02/2017, 0.2 CFU;
    ix. *Smart Nanodevices for Theranostics,* Ilaria Rea, 24/02/2017, 0.3 CFU;
    x. *Fuzzy Logic, Genetic Algorithms and their application to Next Generation Networks,* Leonard Bapolli, 10/03/2017 – 14/02/2017, 0.8 CFU;
    xi. *Dataflow Supercomputing for BigData,* Flora Amato, 12/04/2017, 0.6 CFU.

3. Research Activities

a. Title: *Improving Multibank Memory Access Parallelism on SIMT Architectures.*
b. In the last few years Exascale Computing is constantly gaining importance in the Scientific Community and High-Performance Computers are now considered essential for some experiments in certain areas. For example, in meteorology they are used to create a comprehensive earth system model at 1 Km scale in order to

study cloud convection and ocean eddies, or in biology to simulate entire cells at molecular, genetic, chemical and biological levels. Recently, High-Performance Computers have changed in Heterogeneous Systems. In fact, now it is possible integrate in the same system many core CPUs, accelerators like GPU and FPGA. The main reason of this change is the breakdown of the Dennard Scaling due to leakage currents problem, an electrical phenomenon until then ignored. In fact, since 2005, manufacturers have been forced to switch to multi- and many-core architectures. However, also these architectures have reached a limit in terms of peak performance and increasing it, with the current technology, is hard. In this context, the *Performance per Watt* evaluation metric, that is the rate of computation that can be delivered by a system for every watt of power consumed, is gaining more and more importance, even more than peak performance. Several studies have shown that the value of this metric is often lower than desired, because there is a huge overhead essentially due to *data movement* and *memory accesses*. Just consider that for a single floating-point operation, we have the 85% of overhead in terms of energy consumption. Therefore, my research is focused on memory management and how its improvements can increase the value of this metric. I want to prove the existence of a strong relationship between the data memory allocation, the conflicts generated by threads trying to access data stored in the banks of shared memories and the power consumption. I propose a methodology for exploring conflict-free memory mapping schemes, focusing on a recurrent access pattern in many performance-critical applications, which we called *Transpose-Like*. In this pattern, store operations are performed row-wise while load operations are performed column-wise, or vice versa. Because of the finite number of banks in the local memory, different store/load operations can incur conflicts. Existing programming practices for reducing or avoiding conflicts, like *padding*, involve limited modifications to the code but incurs some memory overhead. The proposed methodology relies on an Integer Linear Programming (ILP) model to describe the problem in terms of linear equalities ensuring optimal bank mapping strategies. All feasible solutions of our ILP model guarantee that there aren't conflicts and memory overhead.

4. Credit Summary

	Credits year 1								Credits year 2							
		1	2	3	4	5	6			1	2	3	4	5	6	
	Estimated	bimonth	bimonth	bimonth	bimonth	bimonth	bimonth	Summary	Estimated	bimonth	bimonth	bimonth	bimonth	bimonth	bimonth	Summary
Modules	20	0	6	3	5	0	9	**23**	10	9	3	0	0	0	6	**18**
Seminars	5	1,8	0,4	0,7	1	0	1,3	**5,2**	5	2,9	1,3	1,4	0	0	0	**5,6**
Research	35	7	4	6	4	7	4	**32**	45	1	5	7	10	10	3	**36**
	60	8,8	10,4	9,7	10,0	7,0	14,3	**60,2**	60	13	9,3	8,4	10	10	9	**60**