

## MULTI-OBJECTIVE OPTIMIZATION FOR AN ENHANCED MULTI-CORE SNIPER SIMULATOR

Radu CHIȘ, Adrian FLOREA, Claudiu BUDULECI, Lucian VINȚAN

“Lucian Blaga” University of Sibiu, Computer Science and Electrical Engineering,  
4 Emil Cioran str., Sibiu, România

Corresponding author: Adrian FLOREA, E-mail: [adrian.florea@ulbsibiu.ro](mailto:adrian.florea@ulbsibiu.ro)

**Abstract.** In this paper we present a four objectives automatic design space exploration methodology for the Sniper multi-core simulator using our multi-objective optimization tool called FADSE. Our target was to search for the quasi-optimal Pareto CPU microarchitectures which simultaneously optimize the following four objectives: integration area, processing performance, energy consumption and thermal behavior. The temperature metric represents the 4<sup>th</sup> objective added by us in the automatic optimization process, which provides for sure a more realistic design space exploration. With the proposed improvements we have found, for each of the objectives, better configurations than the original Gainestown 4 cores microarchitecture configuration used by Sniper. Also, the experimental results showed that the automatic optimization in the 4-objectives space provides better Pareto configurations than our previous methodology, working in a 3-objectives space.

**Key words:** design space exploration, multi-objective pareto optimization algorithms, NSGA-II, processor architecture, sniper multi-core simulator, SPLASH-2 benchmarks, temperature.

### 1. INTRODUCTION

Multi-core and many-core systems represent the de facto standard for today’s processor architectures. However, the actual standard is that the high majority of software programs are still written in sequential programming languages [1]. In other words, the living paradigm is that today’s software is built for past hardware, involving thus a semantic gap. The performance achieved by writing concurrent programs is disregarded, mainly because of the complexity and low productivity imposed by the parallel programming models. All work presented in this paper focuses exclusively on evaluating concurrent software applications.

An open problem for complex computing systems is to find the best micro-architectural configurations or Pareto individuals, which simultaneously fill the needs of different application-specific objectives. Designers are using different simulation based heuristic techniques to tune micro-architectural parameters (cache sizes, number of cores, interconnection networks types, internal core components) to find the best configurations. The time required to find it must be as small as possible. Providing high-end processors, consuming less energy represents a hard constraint also from market competition viewpoint.

The main contribution of this article is to develop an automatic 4-objectives (called 4D) design space exploration (DSE) methodology performing a realistic optimization of the Central Processing Unit (CPU) and finding the best multi-core micro-architectural configurations which simultaneously optimize the four CPU objectives. The targeted objectives are the followings:

- Integration area of the chip must be as small as possible;
- Energy consumption must be as low as possible;
- Processing performance must be as high as possible;
- Temperature of the chip must be as low as possible.

We ran an automatic heuristic search for the best configurations using our previously developed FADSE (Framework for Automatic Design Space Exploration) tool [2] and the resulted micro-architectural configurations were evaluated using state of the art simulators. The Sniper Multi-core (including McPAT [14]), which is de facto standard for next generation processors, developed as a joint project by Intel and

academic researchers [3] was used to compute the following objectives: area of integration, energy consumption and the processing performance. The thermal behavior of the chip is estimated using HotSpot [9], a state of the art thermal modelling simulator that was integrated by us into the Sniper framework.

One of the main scientific gains of this paper consists in exploiting the synergism between Computer Architecture and CPU multi-objective optimization techniques, on one side, and CPU Design Domain Knowledge expressed through some deterministic feasibility rules, on another side, integrated in an automatic 4D DSE tool. This new approach generates better more realistic Pareto CPU configurations compared with all other optimization methods, very useful for commercial implementations.

The rest of this paper is structured as follows. Section 2 describes the related work in the design space exploration field, whereas Section 3 explains our multi-objective optimization methodology. Section 4 illustrates the simulation results while Section 5 concludes the paper suggesting some future work directions.

## 2. RELATED WORK

The research presented in this paper further enlarges and improves our works published in [4] and [8]. Paper [4] presents a multi-objective hardware-software co-optimization for the Sniper multi-core simulator. The targeted objectives were Sniper's intrinsic outputs: area of integration, energy consumption and processing performance. The automatic design space exploration was developed using our previously developed FADSE optimization framework. The authors have varied both hardware (number of cores, cache sizes, cache associativity) and software (GCC optimizations, number of threads and scheduler's strategies) parameters in order to find the quasi-optimal Pareto configurations which simultaneously optimize all the three objectives. This paper proved that optimizing only the hardware related parameters narrows the design space. Better configurations were obtained by simultaneously tuning both hardware and software parameters.

Previously we have enhanced – using a manual approach – the Sniper state of the art multi-core simulator with thermal measurement possibilities using the HotSpot simulator. Paper [8] systematically presents the integration of HotSpot into Sniper simulator. It was implemented a plugin that interacts with Sniper and McPAT in order to generate a power sampling trace and the integration areas for each functional unit of Pareto 3D (three objectives: performance, energy and integration area) optimal microarchitectures found by FADSE. The developed plugin automatically builds the spatial configuration (floorplan) of the simulated microarchitecture. At the end of the simulation the HotSpot thermal estimator is called and the temperature trace over time is generated. Obviously, this Hill Climbing approach does not provide the true optimal Pareto individuals in a 4D objective space. However, as far as we know, through this cited work we were the first researchers that extend Sniper multicore simulator from 3 to 4 objectives, achieved by adding automatic intrinsic temperature measurements and developing an automatic 4D optimization in Computer Architecture field. In contrast with [8], through this paper, by developing automatic 4-objectives DSE we provide an adequate method to obtain a good approximation of the true Pareto individuals.

In [15] the authors are focusing on speeding up the automatic DSE process. The proposed novel solution is named DESPERATE++ and it combines two concepts: a simulation time predictor and an analytical model which provide the quality of a microarchitecture configuration. The simulation time predictor is used to inform if there is enough time to run additional simulations in the reserved time. The selection of the remaining configurations is enforced by the configuration quality predictor. The authors achieved a 4x speedup compared to some state of the art DSE approaches (MOA, NSGA-II and others).

Another useful framework for automatic DSE is Multicube Explorer [16]. It combines the traditional Design of Experiments (DoE) and Response Surface Modelling (RSM) techniques. DoE helps in selecting relevant design points from the whole design space. The RSM tries to find relations between the micro-architectural parameters and the response variables. In this work the targeted objectives are the number of cycles needed by the application to run on the target configuration and the energy consumption (2D).

M3Explorer [16] represents a DSE framework that includes many design space exploration algorithms and can accelerate the DSE process through RSM. Archexplorer [7] is another useful DSE tool where the users can upload their microarchitecture components on a website to be integrated into a computer system simulator. A certain design can be compared against other similar designs introduced by other users. NASA [12] is another effective optimization tool, similar to FADSE because it allows the user to connect their DSE

designs to any simulator. In contrast with our developed multi-objective Pareto optimization methodology, it performs an optimization using single objective genetic algorithms for each objective (1D). Magellan [13] implements a DSE tool which is bounded to a certain simulator (SMTSIM) and can perform only single objective optimizations, too.

In [11] the FADSE tool was used to explore the vast design space of the Grid ALU Processor (GAP) and its post-link optimizer (GAPtimize). FADSE proved able to find an approximation of the Pareto [2] frontier consisting of near-optimal individuals in a feasible time. To our knowledge, the only multi-objective optimization tool that accelerates and improves the DSE process through a domain-knowledge, represented by fuzzy logic rules written in a human-readable form and deterministic constraints, is FADSE.

In [17] the authors are focusing on developing a DSE method for mapping of concurrent application tasks onto architectural resources of an MPSoC. Domain knowledge is used to guide the genetic algorithm towards better results by removing the symmetry from the design space and by using a new effective crossover operator. In contrast to our work, their domain knowledge is not focused on finding the best parameters values for multi-core hardware architecture from a multi-objective point of view.

In this work the main aim is to optimize our extended multi-core Sniper simulator having 4 intrinsic objectives (integration area, energy consumption, performance and thermal behavior) through automatic design space exploration using our previously developed FADSE tool.

### 3. SIMULATION METHODOLOGY

This section presents the simulation methodology used to find the quasi-optimal Sniper multi-core configurations, from the huge design space (352.800 configurations) that exists. The problem solved in this paper is a 4-objective minimization using FADSE for the enhanced Sniper simulator that automatically generates the following outputs: temperature, integration area, energy consumption and number of cycles per instructions (for performance; CPI) and it is called by us “4D optimization”.

An overview of the simulation methodology is presented in Fig. 1. On the left side a global overview of the FADSE tool can be observed. The Framework for ADSE (Automatic Design Space Exploration) is configurable via three input files (framework, clients and meta-heuristic configuration) and it is composed of a server and multiple clients. Briefly, the server has the role of distributing simulations among clients, centralizing the results and conducting the design space exploration process. The clients perform the actual simulations of the individuals in parallel and send the results back to the server.

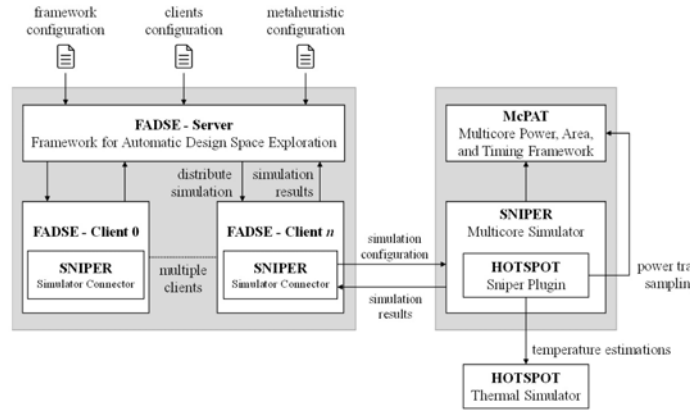


Fig. 1 – FADSE in 4D simulation methodology overview.

For the actual simulations the FADSE clients call the Sniper simulator through a special connector which was introduced in [4]. The connector does the bidirectional translation of the configuration parameters used by the heuristic algorithm in FADSE and the actual parameters which are used by the Sniper command line. Furthermore, the connector computes the objectives using the results provided by Sniper. For the temperature objective the maximum value founded inside the temperature trace is taken. The temperature trace is estimated by the thermal simulator HotSpot. The integration area and the performance values are returned exactly as McPAT and Sniper computes them. The energy consumption is calculated with Eq. (1):

$$E[\text{Joules}] = \frac{P_{\text{total}}[\text{W}]}{f_{\text{CPU}}[\text{Hz}]}, \quad (1)$$

where:  $P_{\text{total}}$  [W]: represents the total power consumption (provided by McPAT);  $f_{\text{CPU}}$  [Hz]: represents the frequency of the simulated microprocessor.

The microprocessor optimized in this work and simulated by the Sniper multi-core simulator is produced by the Intel company and it is named “5500-series - Gainestown” or “Nehalem-EP” [10].

Table 1

Gainestown CPU characteristics [10]

Production date	Since 2008 to present
CPU frequency range	From 1.86 GHz up to 3.33 GHz
Integration technology scale	45nm
Instruction set	x86
Microarchitecture	Nehalem
Cores	4
L2 cache	4×256 kB
L3 cache	8 MB
Package(s)	LGA 1366
Brand name(s)	Xeon 55xx

Towards finding the quasi-optimal configuration “hidden” among the 352.800 configurations of the full search space, the hardware related parameters from Table 2 were varied. All parameters are represented as two’s exponent inside FADSE. For example, the number of cores can range from 1 to 16 according to Table 2. The actual range will be  $2^0$ ,  $2^1$ ,  $2^2$ ,  $2^3$  and  $2^4$ , in total there are 5 distinct values. Although the design space seems quite small, evaluating one configuration on the small input size of the shortest benchmark (*radix*) from the SPLASH-2 [18] suite takes 1 minute using an Intel Quad Core I7, 4.4 GHz, 16 GB DRAM host machine. Computing all these configurations on such a machine would take more than 250 days.

Table 2

Hardware related parameters

Parameter	Values		
	Min	Max	Distinct
Number of cores	1	16	5
DRAM interleaving controllers	1	64	7
L1 Data Cache Associativity	8	16	2
L1 Data Cache Size [KB]	32	256	4
L2 Cache Associativity	8	16	2
L2 Cache Size [KB]	32	2048	7
L3 Cache Associativity	8	16	2
L3 Cache Size [KB]	128	32768	9
L3 Cache Shared Cores	1	16	5

To get more accurate results, simulation on the large input size with all SPLASH-2 benchmarks are needed, which would take at least two orders of magnitude longer to simulate. Our results should be regarded as a proof-of-concept, that the run-time floorplan generation and temperature estimations can be automatically computed using the FADSE tool and the Sniper multi-core simulator. The selected state of the art meta-heuristic for the multi-objective DSE is NSGA-II [6] (Non-dominated Sorting Genetic Algorithm). The configuration parameters for the search algorithm are presented in Table 3.

Table 3

NSGA-II characteristics

Crossover operator	Single point
Crossover probability	90%
Mutation operator	Bit-flip
Mutation probability	16%
Selection operator	Binary tournament
Population size	50 individuals
Stop condition	Stops after 50 generations

We ran the DSE process over two runs, with 3 and 4 objectives respectively, for 50 generations, each generation consisting of 50 individuals. Each individual was simulated over 4 distinct benchmarks belonging to the SPLASH-2 suite. The used benchmarks are the followings: *fft*, *radix*, *lu.cont*, *ocean.cont* and they were run with the input size small. To eliminate some of the unfeasible configurations which may result during the automatic design space process the following deterministic feasibility rules were introduced:

- L2 cache size > L1 data cache size;
- L3 cache size > L2 cache size.

The individuals which violate at least one of the above rules are marked as unfeasible and a new individual is automatically generated to replace the problematic individual. This situation can happen during the ADSE process at some points after offsprings development or after the initial population generation.

For thermal estimations the power consumption sampling is done at 500 ms, the simulation model used is block and the heat sink was adjusted exactly as presented in [8]. The selected cooling package consists of a simple heat sink which sits on top of the chip, without forced air convection (no air fan).

Three FADSE clients were used in parallel in order to accelerate the automatic DSE process. The used version of the Sniper simulators was 6.1 and for the HotSpot simulator version 5.02. The total simulation time was around 2 weeks and about 1% of the total individuals were evaluated. FADSE is a reliable tool able to cope with failing clients, failing networks or even power loss of the entire system. Using the checkpointing mechanism, FADSE is able to recover from these situations by detecting the problems and resubmitting the simulations to other clients. For accelerating the DSE process and reduce the simulation time, our DSE tool stores and reuses, through a dedicated database, the already simulated individuals [2].

#### 4. QUALITATIVE AND QUANTITATIVE RESULTS. INTERPRETATIONS

First, we compared the two runs, 4 objectives (including temperatures) vs. 3 objectives and corresponding temperatures computed afterwards, using the coverage metric, which compares the fraction [%] of individuals from one run that are non-dominated by the individuals from the other run. Our results show that each of the runs has around 20% of the individuals non-dominated by individuals from the other run. Because the coverage takes into consideration just the number of the individuals that are dominated, it can give us some false impressions.

Exclusively looking at Fig. 2, we could interpret the results saying that the two runs are almost equally powerful, being better and worse on different segments of the generations. From the 10<sup>th</sup> to the 35<sup>th</sup> generation, the run with 4 objectives has more non-dominated individuals, meaning that for up to 45% of the total or 22 individuals in the 20<sup>th</sup> generation, there are no configurations from the 3 objectives run that are better in respect to any objective.

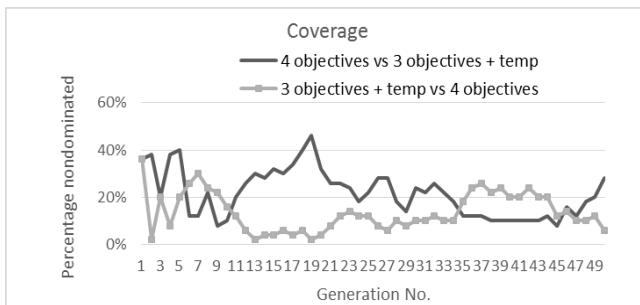


Fig. 2 – Coverage comparison.

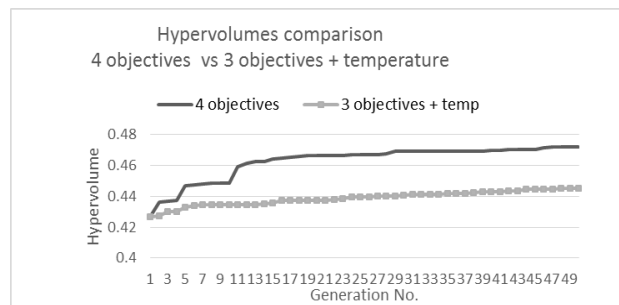


Fig. 3 – Hypervolumes comparison.

Looking now at Fig. 3, it can be seen that the coverage could have misled us and that the number of individuals that are dominated/non-dominated are less relevant in the quality of the runs. It can be seen that the hypervolume of the 4 objectives run converges faster and also yields better results, which is remarkable. Both runs seem to improve the quality of the results for the 50 simulated generations, meaning that, in theory, we could find better results if we were to simulate further.

The hypervolume metric has shown us that the 4 objectives run is better than running with just 3 objectives and making the temperature computations after, but Fig. 3 cannot show if the hypervolumes coincide or dominate different parts of the objectives space. This is where the Two Set Hypervolume Difference (TSHD) metric is very useful.

Looking at the TSHD in Fig. 4, it can be seen that about 95% of the hypervolume is common and that the two different runs contribute quite differently to the overall solutions quality.

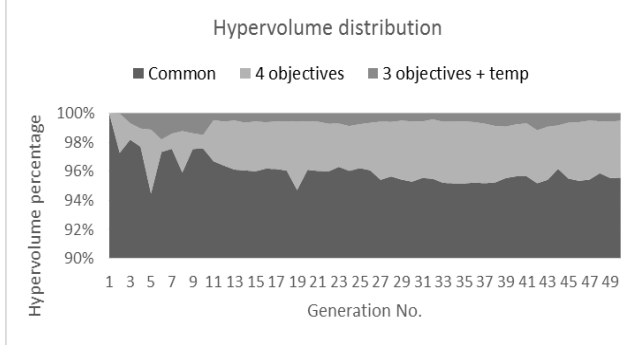


Fig. 4 – Two set hypervolume difference.

Figure 4 shows that the hypervolume dominated only by the 4 objectives run represents ~ 4% of the total hypervolume and seems to increase over the 50 generations, while the hypervolume dominated only by the 3 objectives run stagnates at ~ 1%.

Table 4 shows some of the best found individuals with regards to the 4 objectives and the results are according to the designer's intuition. It can be seen that the best Area, Energy and Temperature correspond to only one core having small cache sizes, while the best found in terms of the CPI corresponds to 16 cores and the highest possible cache sizes.

The difference between the maximums of the best and worst temperature is about 15–16 Celsius degrees on our simulated benchmarks. The Pareto individuals generated by our developed 4 objectives automatic methodology are better than the individuals generated through our previous developed 3 objectives automatic methodology followed by temperature calculation (see Fig. 4).

Table 4

Best found Pareto individuals

	Best Area	Best Energy	Best CPI	Best Temp
Number of cores	1	1	16	1
DRAM interleaving controllers	2	2	2	16
L1 Data Cache Associativity	8	8	8	8
L1 Data Cache Size [KB]	32	32	256	64
L2 Cache Associativity	16	16	16	8
L2 Cache Size [KB]	256	256	2048	256
L3 Cache Associativity	16	16	16	16
L3 Cache Size [KB]	1024	1024	32768	32768
L3 Cache Shared Cores	1	1	2	1
Area	35.67	35.67	2957.61	280.21
Energy	6.01E-09	6.01E-09	1.08E-07	1.06E-08
CPI	1.22	1.22	0.18	1.04
Temperature	54.41	54.41	67.39	51.66

Table 5

3 obj. + T vs. 4 obj. same number of cores comparison

	3 obj. + T	4 obj.	Gain [%]	3 obj. + T	4 obj.	Gain [%]
Number of cores	2	2		8	8	
DRAM interleaving controllers	16	2		8	2	
L1 Data Cache Associativity	32	128		32	32	
L1 Data Cache Size [KB]	8	8		8	8	
L2 Cache Associativity	128	256		128	256	
L2 Cache Size [KB]	16	16		16	16	
L3 Cache Associativity	8192	8192		8192	1024	
L3 Cache Size [KB]	16	16		16	16	
L3 Cache Shared Cores	1	1		4	1	
Area	186.01	189.17	-1.70	376.54	264.82	29.67
Energy	1.28E-08	1.18E-08	8.42	2.77E-08	2.75E-08	0.78
CPI	0.577	0.571	1.02	0.243	0.235	3.37
Temperature	57.66	55.46	3.82	63.59	57.24	9.99

Also looking at some specific Pareto individuals in Table 5, we can see that for the same number of cores, e.g. 2 and 8, the 4 objective run finds even non-dominated individuals that are much better than the 3 objective run + temperature. They are feasible individuals for commercial hardware implementations having equilibrated effective values for all the 4 objectives. In these cases, some subtle non-intuitive inter-related DRAM and cache design characteristics made the difference. It would be quite impossible for the designer to discover such optimal multicore systems based on intuition. More than this, even the 3 objectives automatic methodology followed by temperature calculation missed such very effective multicore systems.

Table 6

Average simulation time results collected from Splash-2 benchmarks suite  
small input size for 1 individual

# cores	$T_{4D} = T_{3D} + T_H$	$T_H$	$T_{3D}$
1	980 s	97 s	883 s
2	754 s	99 s	655 s
4	652 s	111 s	541 s
8	607 s	147 s	460 s
16	799 s	268 s	531 s
Average time →	<b>758 s</b>	<b>144 s</b>	<b>614 s</b>

Below we present some considerations related to the time overhead of finding the best configurations in our 4D approach comparing with the previous developed 3D approach. For proper understanding we make the following notations:

$T_{3D}$  – average time (measured on 1, 2, 4, 8 and 16 cores) required for evaluation of an individual from 3 objectives point of view (Energy, CPI, Area). In our case,  $T_{3D} = 614s \cong 10.23$  minutes (see Table 6).

$T_H$  – average time (measured on 1, 2, 4, 8 and 16 cores) required for running Hotspot simulator for an individual. In our case,  $T_H = 144s \cong 2.4$  minutes (see Table 6).

$T_{4D}$  – average time (measured on 1, 2, 4, 8 and 16 cores) required for evaluation of an individual from 4 objectives point of view (Energy, CPI, Area, Temperature). In our case,  $T_{4D} = 758s \cong 12.63$  minutes.

$G$  – number of generations. In our case,  $G = 50$  (see Table 3).

$I$  – number of individuals evaluated. In our case, using the NSGA-II, it is twice the population size ( $I = 2 \times 50 = 100$ , combining parent and children solutions – see Table 3).

We denoted  $T_S$  as time to set the dominance of individuals from NSGA-II selection process. It depends by the number of objectives but its value is rather small. We take an average measured value of 10 minutes.

$Total_{4D}$  = total simulation time for all  $G$  generation from 4D objective evaluation. It roughly consists in the **fitness evaluation of each individual** ( $T_{4D}$ ) and after that **setting the dominance** ( $T_S$ ).

$Total_{3D}$  = total simulation time for all  $G$  generation from 3D objective evaluation ( $T_{3D}$ ), setting the dominance ( $T_S$ ) and followed by temperature evaluation only of configurations situated on Pareto Front ( $T_H$ ).

$$Total_{4D} = G \cdot (I \cdot T_{4D} + T_{S_{4objectives}}), \quad (2)$$

$$Total_{3D} = G \cdot (I \cdot T_{3D} + T_{S_{3objectives}}) + \frac{I}{2} \cdot T_H, \quad (3)$$

$$Speed_{up} = \frac{Total_{4D}}{Total_{3D}} = \frac{G \cdot (I \cdot T_{4D} + T_{S_{4objectives}})}{G \cdot (I \cdot T_{3D} + T_{S_{3objectives}}) + \frac{I}{2} \cdot T_H}. \quad (4)$$

Replacing variables with their values in Eq. 4, the time overhead of finding the best configurations using an automatic 4-objectives design space exploration methodology is about 23%. However, it provides significantly better Pareto configurations than the previous methodology (3D optimization, followed by manual computation of the 4th objective afterwards). However, this overhead might decrease if the number of cores of the host architecture increases, favoring the parallel evaluation of individuals from population.

## 5. CONCLUSIONS AND FURTHER WORK

It has been showed that the new extension of the Sniper multi-core simulator used by FADSE can yield better realistic results for 4 objectives than before. More precisely, the experimental results showed that the automatic 4D optimization DSE process provides significantly better Pareto configurations than the previous DSE methodology (run with only 3 objectives and the computation of the 4-th objective afterwards), considering the proposed multi-objective approach. According to our knowledge, we are the first researchers developing an automatic 4D optimization for a complex multicore system. Thus, we contribute to a more realistic CPU optimization process. The biggest advantage is the run-time generation of the floorplan, temperature computation and the possibility to run a DSE process automatically. The TSHD metric convincingly shows the power of our new approach. The hypervolume non-dominated by the new run is 4 times greater than the hypervolume non-dominated by the 3 objectives run. The differences in temperature between the best and worst maximum are about  $\sim 15\text{--}16$  °C. Due to the implementation's nature of the Sniper extension, our approach can be reused by other researchers using other DSE tools, which wish to automatically compute floorplans and temperatures. All the previous configurations for Sniper and Hotspot still work and our extension can be switched on or off through a command line parameter.

A first straightforward further work idea is to extend this research by developing an optimization method for both hardware and software parameters. Also, we plan to accelerate the DSE process by creating a RSM for Sniper and HotSpot simulators to reduce the evaluation time. In order to do this, we will not use the classical equations (polynomial, etc.) tuning approach. Instead, we will use a knowledge discovery approach using genetic programming [5] to find the best fitting equations. We also plan to restrain the size of the search space by creating a domain-ontology for Sniper multicore expressed using fuzzy logic rules and other knowledge representation methods.

## REFERENCES

- BRIDGES M., VACHHARAJANI N., ZHANG Y., JABLIN T., AUGUST D. *Revisiting the Sequential Programming Model for Multi-Core*, Proceedings of the 40<sup>th</sup> Annual IEEE/ACM International Symposium on Microarchitecture (MICRO 40), pp. 69–84, IEEE Computer Society, 2007.
- CALBOREAN H., *Multi-Objective Optimization of Advanced Computer Architectures using Domain-Knowledge*, PhD Thesis, “Lucian Blaga” University of Sibiu, 2011.
- CARLSON T.E., HEIRMAN W., EECKHOUT L., *Sniper: Exploring the Level of Abstraction for Scalable and Accurate*, International Conference for High Performance Computing Networking, 2011, pp. 52:1–52:12.
- CHIȘ R., VINȚAN L. *Multi-Objective Hardware-Software Co-Optimization for the SNIPER Multi-Core Simulator*, Proceedings of 10<sup>th</sup> International Conference on Intelligent Computer Communication and Processing, Cluj-Napoca, 2014.
- COOK H., SKADRON K., *Predictive Design Space Exploration Using Genetically Programmed Response Surfaces*, Proceedings of the 45<sup>th</sup> Annual Design Automation Conference, 2008.
- DEB K., PRATAP A., AGARWAL S., MEYARIVAN T., *A fast and elitist multiobjective genetic algorithm: NSGA-II*, IEEE Transactions on Evolutionary Computation, 6, 2, 2002.
- DESMET V., GIRBAL S., TEMAM O., FRANCE B.F., *Archexplorer.org: Joint compiler/hardware exploration for fair comparison of architectures*, Proceedings of the 6<sup>th</sup> HiPEAC Industrial Workshop, Paris (France), Nov. 2008.
- FLOREA A., BUDULECI C.R., CHIȘ R., GELLÉRT A., VINȚAN L., *Enhancing the Sniper Simulator with Thermal Measurement*, The 18<sup>th</sup> International Conference on System Theory, Control and Computing, Sinaia, 2014, pp. 31–36.
- HUANG W., SKADRON K., GURUMURTHI S., RIBANDO R.J., STAN M.R., *Differentiating the Roles of IR Measurement and Simulation for Power and Temperature-Aware Design*, IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), 2009.
- Intel Corporation. *Optimizing Intel® Xeon® Processor 5500 Series (Nehalem-EP Processor) Based Workstation and Server Platforms for the ENERGY STAR\* Program*, Document Number 414863 Revision 2.0, 2009.
- JAHR R., UNGERER T., CALBOREAN H., VINȚAN L., *Automatic Multi-Objective Optimization of Parameters for Hardware and Code Optimizations*, Proceedings of the International Conference on High Performance Computing & Simulation, Istanbul (Turkey), July 2011, pp. 308–316.
- JIA Z.J., PIMENTEL A.D., THOMPSON M., BAUTISTA T., NÚÑEZ A. *NASA: A generic infrastructure for system-level MP-SoC design space exploration*, 8<sup>th</sup> IEEE Workshop on Embedded Systems for Real-Time Multimedia, pp. 41–50, 2010.
- KANG S., KUMAR R., *Magellan: a search and machine learning-based framework for fast multi-core design space exploration and optimization*, Proceedings of the conference on design, automation and test in Europe, Munich (Germany), pp. 1432–1437, 2008.
- LI S., AHN J.H., STRONG R., BROCKMAN J., TULLSEN D., JOUPPI N., *McPAT: An integrated power, area, and timing modeling framework for multicore and manycore architectures*, Microarchitecture (MICRO-42), 2009 (Romania); 42<sup>nd</sup> Annual IEEE/ACM International Symposium, 2009, pp. 469–480.



15. MARIANI G., PALERMO G., ZACCARIA V., SILVANO C. *DeSpErate++: An Enhanced Design Space*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, **34**, 2, pp. 293–306, 2015.
16. SILVANO C., FORNACIARI W., PALERMO G., ZACCARIA V., *MULTICUBE: Multi-Objective Design Space Exploration of Multi-Core Architectures*, Proceedings of the IEEE Annual Symposium on VLSI, Greece, 2010, pp. 488–493.
17. THOMPSON M., PIMENTEL A.D., *Exploiting domain knowledge in system-level MPSoC design space exploration*, Journal of Systems Architecture, **59**, 7, pp. 351–360, 2013.
18. WOO S.C., OHARA M., TORRIE E., SINGH J.P., GUPTA A., *The SPLASH-2 Programs: Characterization and Methodological Considerations*, 22<sup>nd</sup> International Symposium on Computer Architecture, 1995.
19. ZACCARIA V., PALERMO G., MARIANI G., CASTRO F., SILVANO C., *Multicube Explorer: An Open Source Framework for Design Space Exploration of Chip Multi-Processors*, Workshop on Parallel Programming and Run-time Management Techniques for Many-core Architectures (2PARMA), 2010.

Received Octobre 2, 2017