LBUS Framework for Multi-Objective Optimization using Domain-Knowledge

The main aim of this presentation is to point out the actual Computer Architecture research developed by The Advanced Computer Architecture and Processing Systems (ACAPS) Research Centre from "Lucian Blaga" University of Sibiu (LBUS), Romania – see http://acaps.ulbsibiu.ro/index.php/en/. Since 2012 I am a HiPEAC member and some of my ACAPS collaborators are HiPEAC affiliated members, too.

The main aim of our present-day Computer Architecture research consists in developing innovative computer architectures and optimizing them. Some of the subsequent research objectives are the followings:

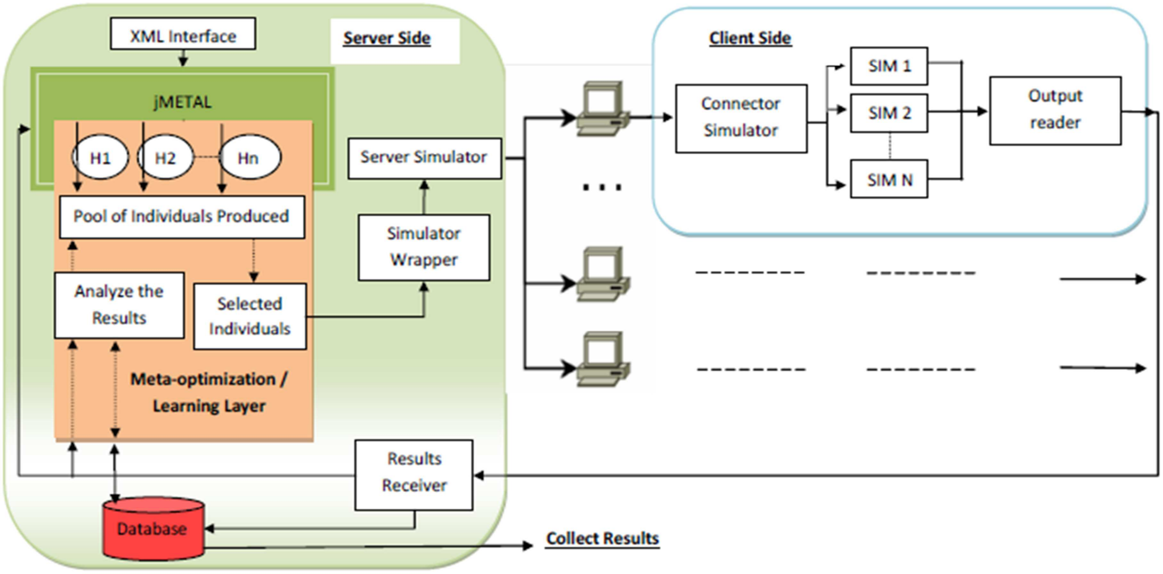
- Developing some novel effective micro-architectures (microarchitectures with predictive-speculative processing, Network on Chip, etc.)
- Develop a robust and fast automatic design space exploration framework for *hardware-software* optimization of complex computer systems
- Research how domain-knowledge could be represented and integrated into the *Design Space Exploration* (DSE) algorithms
- Quantify domain-knowledge impact on the DSE process
- Evaluate & compare different multi-objective DSE algorithms
- Meta-optimization (adaptive selection of DSE algorithms), etc.

Multi-objective optimization (performance, power consumption, temperatures, complexity…) of computing systems having many parameters is for sure a very complex problem. Not only the hardware needs to be simulated and evaluated; usually we need hardware and software co-optimization (cross-layer optimization). Usually exhaustive search is prohibited due to the enormous design space. The solution consists in developing and implementing some advanced heuristic algorithms in order to solve this problem. In our optimization research we used a Pareto-based approach by implementing some multi-objective evolutionary (genetic) algorithms and bio-inspired algorithms belonging to the Particle Swarm Optimization class. The solutions' quality is usually evaluated through some implemented metrics like hyper-volume (also used as a stop condition), 7 point average distance, coverage (for comparing two DSE algorithms), Two Set Hyper-volume Difference (for better understanding about how much one DSE algorithm dominates the other one in a multi - dimensional space), etc.

All these briefly presented DSE characteristics were implemented by us in a dedicated software product entitled *Framework for Automatic Design Space Exploration* (*FADSE*). It was initially designed and implemented by my former PhD student Horia Calborean under my scientific coordination and it is further developed by the ACAPS research team. The software tool is publicly available (see https://code.google.com/p/fadse/). It includes many state-of-the-art multi-objective DSE algorithms (integrated with *jMetal* library). FADSE can be connected to any existing (architectural) simulator (connector needed). It is easy to use XML configuration interface. It was developed an easy to use interface for connectors to the computer simulators.

The search time is reduced through a database integration (we achieved up to 67% simulations results' reuse). The DSE multi-objective implemented evolutionary algorithms were adapted in order to allow distributed evaluations on LANs or High Performance

Computers. Also FADSE implements some reliability techniques: if clients do not respond, networks fail, the simulation is resubmitted to another client. For problems like: power loss or if the server stops responding, it was implemented a check pointing mechanism. This allows us to restart the DSE process from an intermediate state.

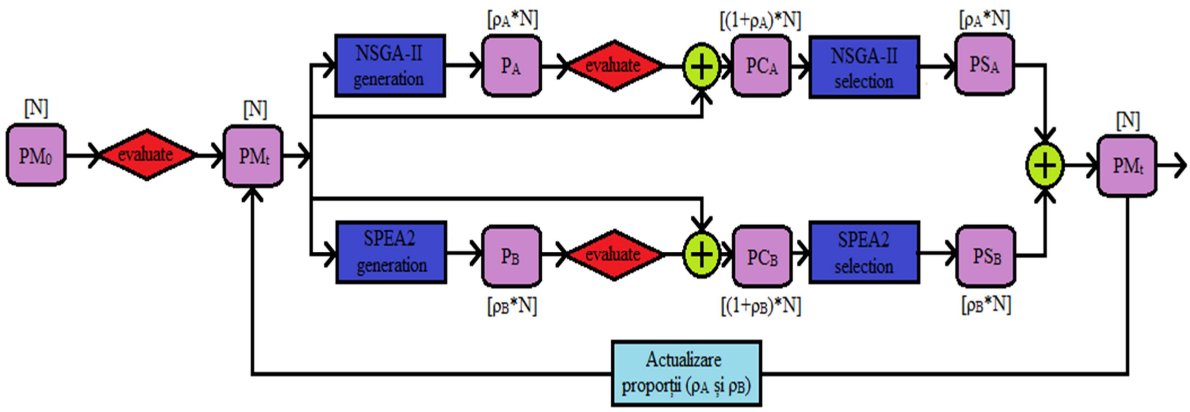


Structure of FADSE (distributed version)

In order to achieve better convergence speed and solutions' quality we implemented specific domain-knowledge for each of the target computer architecture to be optimized. Domain-knowledge is represented by a system of complete, non-redundant and non-contradictory rules or other specific restrictions. As far as we know, we are the first ones using fuzzy logic as a method to express computer architecture knowledge into a DSE tool (Some CPU Fuzzy Logic Rules Examples: IF IL1Cache_Size IS *small* AND DL1Cache_Size IS *small* THEN UL2Cache_size IS *big*, IF IL1Cache_Size IS *big* AND DL1Cache_Size IS *big* THEN UL2Cache_size IS *small*, etc. Particularly, calculating degrees of contradiction between fuzzy logic rules is another scientific objective for us.) For each of the fuzzy logic rule a Fuzzification→ Inference→ Defuzzification (Crisp Value) process is computed. As a consequence, Mutation Genetic Operator in our implemented DSE algorithms was essentially modified as follows:

1. *For all the parameters (genes) in the individual (chromosome);*
   1.1. *If a fuzzy rule exists for the current parameter, having it as a consequent;*
      1.1.1. *Compute COG of this parameter taking into consideration the current values of the other parameters;*
      1.1.2. *Compute the membership $\mu(COG)$ value of the COG;*
      1.1.3. *Generate a pseudo-random number between 0 and 1;*
      1.1.4. *If the previously generated pseudo-random number is smaller than "fuzzy probability";*
         1.1.4.1. *Current parameter is set to a value equal with COG;*
      1.1.5. *Jump to next iteration;*
   1.2. *Otherwise (do bit flip mutation);*
      1.2.1. *Generate a pseudo-random number between 0 and 1;*
      1.2.2. *If the previously generated pseudo-random number is smaller than the probability of mutation;*
         1.2.2.1. *Change the current parameter to a random value;*
      1.2.3. *Jump to next iteration;*
2. *STOP.*

Based on our experience in Computer Architecture multi-objective optimization, there is not a general optimal multi-objective DSE algorithm (one algorithm might converge fastest and other one provides best solution for a certain architecture; for other one might be different). As a consequence, we implemented a new abstraction level in FADSE called meta-optimization. It acts over the domain-knowledge and DSE algorithms (meta-heuristic) levels. Meta-optimization search within a search space of heuristic methods. Meta-optimizations are concerned with intelligently choosing the right heuristic DSE algorithm in a given situation in order to handle a wide range of problem domains rather than current meta-heuristic (DSE algorithm). Developing some effective adaptive meta-optimization algorithms is an important scientific challenge for us. According to our first experiences meta-optimization is used to optimize the performance of design space explorations, driving two different multi-objective DSE algorithms concurrently. More precisely, we selected two genetic algorithms, NSGA-II and SPEA2. In this connection, we developed an elitist meta-optimization algorithm. The algorithm's flow is presented in the next figure. With the proposed improvements, as a first experiment, we ran FADSE in order to optimize the performance parameters of the Grid ALU Processor (GAP) micro-architecture.



Adaptive Met-Optimization Algorithm

We already successfully optimized with FADSE some computer simulators like GAP/GAPtimize (developed at Augsburg University by Professor Theo Ungerer's research group), M-Sim 2 simulator improved by us with a Selective Load Value Predictor, UniMap (our developed NoC simulator), M-Sim 3 multicore simulator, Sniper multicore simulator (with objectives Energy, CPI, Integration Area), etc. For the first three projects we already developed and implemented specific effective domain-knowledge in FADSE. Using them, both the solutions' quality and the convergence speed were improved. Now we are thinking about some multicore systems domain-knowledge (for example related to Sniper Multicore Simulator). We already integrated Hotspot simulator for temperature computations in Sniper simulator. This allows us a 4D space optimization for Sniper Multicore Simulator rather than a 3D space optimization as it is only possible with the standard Sniper simulator. We also successfully used FADSE for optimization of a motor in the acceleration pedal of a car based on a model implemented in Comsol with Matlab, developed by an automotive company; we looked for the biggest constant force.

Some of our published papers focused on multi-objective optimization might be available at http://webspace.ulbsibiu.ro/lucian.vintan/html/#10. More precisely, they are available through the following links:

http://webspace.ulbsibiu.ro/lucian.vintan/html/Date_2010.pdf

http://webspace.ulbsibiu.ro/lucian.vintan/html/CSCS18.pdf

http://webspace.ulbsibiu.ro/lucian.vintan/html/HPCS_2011.pdf

http://webspace.ulbsibiu.ro/lucian.vintan/html/CDT2012.pdf

http://webspace.ulbsibiu.ro/lucian.vintan/html/LNCS2012.pdf

http://webspace.ulbsibiu.ro/lucian.vintan/html/Micro_2013.pdf

http://webspace.ulbsibiu.ro/lucian.vintan/html/CCPE.pdf

As a HiPEAC member our research group is very motivated to develop a useful activity, including scientific collaborations, in the HiPEAC-3 NoE program. We already successfully organized the first HiPEAC Workshop held in Romania, at LBUS, during the period April 1-3 2013. At this workshop participated Prof. Koen de Bosschere – the HiPEAC coordinator from Ghent Univ., Prof. Rainer Leupers – Aachen Univ., Prof. Kostas Magoutis - FORTH Greece and Romanian Professors Nicolae Țăpuș (UP Bucharest), Sergiu Nedevschi (UT Cluj-Napoca), Lucian Vintan, Daniel Volovici, and the doctors Adrian Florea, Arpad Gellert, Horia Calborean and Radu Ciprian (LBUS), Dr. Alexandru Amaricai and Dr. Marius Marcu (UP Timișoara). **Additionally we are especially interested in developing some joint research projects with HiPEAC members in the EC Horizon 2020 Frame.**

Lucian Vintan
"Lucian Blaga" University of Sibiu
E-mail: lucian.vintan@ulbsibiu.ro
http://webspace.ulbsibiu.ro/lucian.vintan/html/