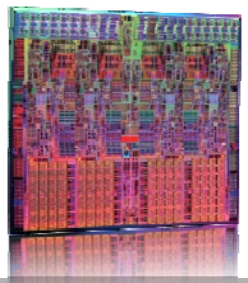
Automatic Design Space Exploration for Chip-Multi Processors

Cristina Silvano
Politecnico di Milano, Milano (ITALY)
Dipartimento di Elettronica e Informazione
cristina.silvano@polimi.it
http://home.dei.polimi.it/silvano/

Outline

- Introduction and Motivations
- Automatic Design Space Exploration Methodology
- MULTICUBE Explorer Tool
- Experimental Results
- Conclusions



Introduction and Motivations

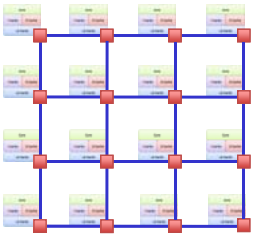
The design space

- In the age of multi/many-core, a wide range of architecture parameters must be tuned to find the best system configuration.
- **Design space** of the target architecture A should consider all possible configurations of each parameters p_i :

$$A = S_{p1} \times S_{p2} \times .... \times S_{pn}$$

- Example:

Parameter	Min.	Max.
# Processors	2	16
Processor issue width.	1	8
L1 instruction cache size	2K	16K
L1 data cache size	2K	16K
L2 private cache size	32K	256K
L1 instruction cache assoc.	1w	8w
L1 data cache assoc.	1w	8w
L2 private cache assoc.	1w	8w
I/D/L2 block size	16	32



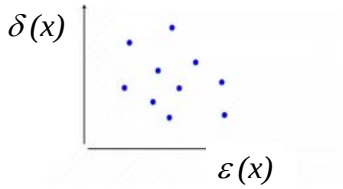
⇒ **Large design space composed of 2^{17} (131 072) system configurations**

The multi-objective optimisation problem

- Objective function: To minimize both energy $\varepsilon(x)$ and execution time $\delta(x)$ of the target application on system configurations x:

$$\min_{x \in X} \omega(x),\ \omega(x) = \begin{bmatrix} \epsilon(x) \\ \delta(x) \end{bmatrix}$$

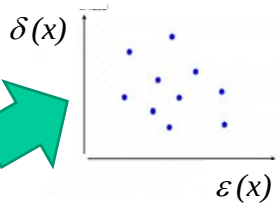
  where X is the design space.
- The solution is a set of tradeoff configurations $X_p \subseteq X$ known as Pareto set



The concepts behind the automatic DSE

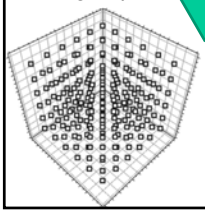
Output Variables define the objective space

The black box generates the output values accordingly to the inputs.



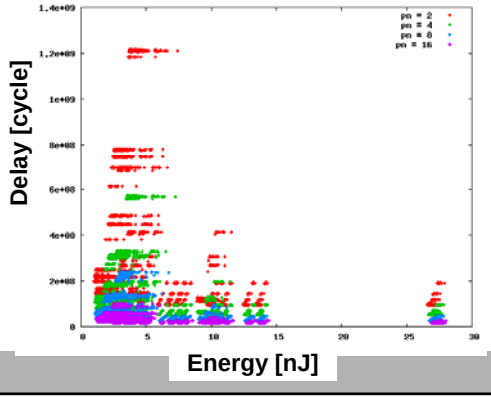
The black box can be:
1) A simulator that models the system behavior and generates output values
2) A set of solvers that models the system behavior and estimates output values

Input Variables: architecture parameters that define the design space



Full Search Design Space Exploration

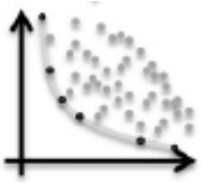
- In most cases, the design space to be explored is **large** and we need to find the best trade-off in terms of **multiple competing objectives**.
- **Full-search exploration** is unfeasible because it requires a very long simulation time
- Example: Design space composed of 2^{17} = 131 072 system configurations. If simulation of the target application for each configuration requires 5 min ⇒ ~ 3 months on 5 parallel machines for full-search exploration of the target application



9

Introduction and Motivation

- Given the increasing complexity of **Chip Multi-Processors**, a wide range of **architecture parameters** must be explored to find the best trade-off in terms of **multiple objectives** (energy, delay, bandwidth, area, etc.)
- **Multi-Objective Exploration** of the huge design space of next generation CMPs cannot be anymore a manual optimisation process based on intuition and past experience of the designer
- **Need for Automatic Design Space Exploration** to support systematically the exploration and the quantitative comparison in terms of multiple competing objectives **(trade-offs analysis)**



10

Motivations

- Why Automatic Design Space Exploration?

  1. Faster exploration time

  2. Better quality of results

Automatic Design Space Exploration

MULTICUBE Project

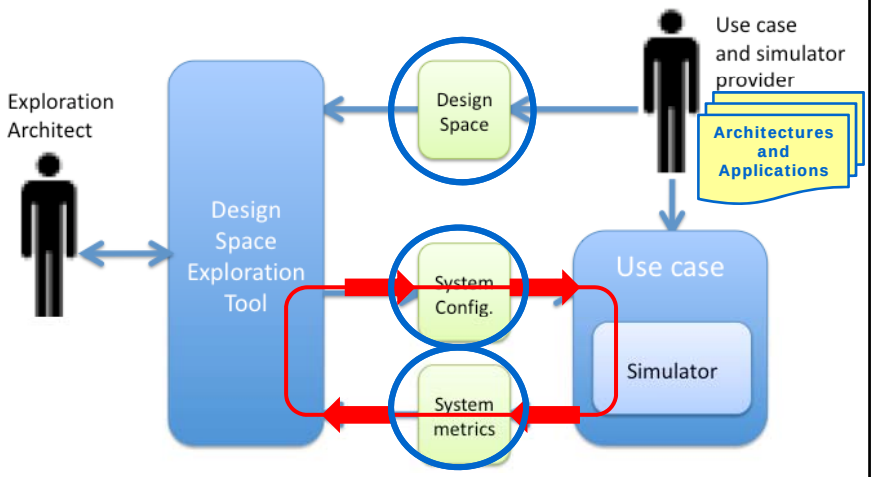
- An **overall design space exploration framework** is needed to combine simulation and optimization techniques into a global search space with a common interface to the simulation and optimisation tools.
- **MULTICUBE FP7-ICT Project** focuses on the definition of an **automatic multi-objective Design Space Exploration (DSE) framework** to be used to tune Chip Multi-Processor architectures evaluating a set of metrics (such as energy and delay) for the next generation embedded computing platforms.

www.multicube.eu

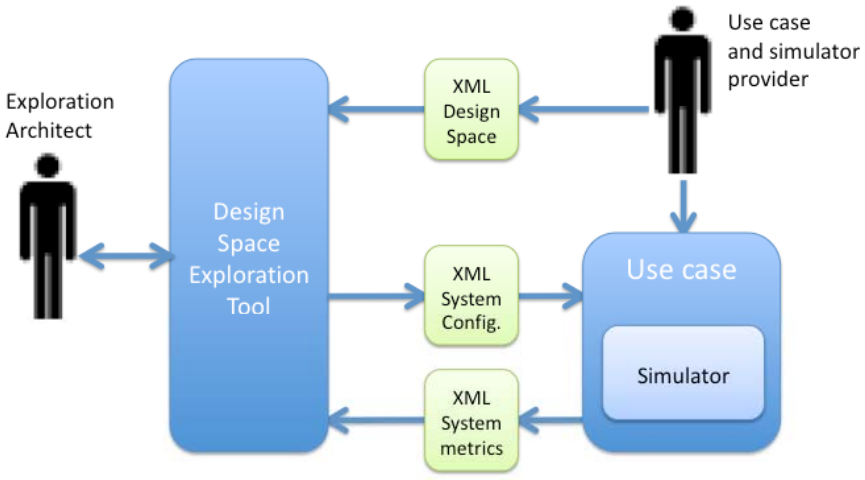
MULTICUBE Explorer: *What is it?*

- MULTICUBE Explorer is an **Open Source Multi-Objective Design Space Exploration (DSE) framework** to tune MP-SoC architectures
- **Efficiency:** MULTICUBE Explorer is an automation and acceleration infrastructure to minimize the numbers of simulations to be executed during the DSE process
- **Flexibility:** MULTICUBE Explorer is a design optimization and exploration infrastructure where you can easily plug-in your own simulator and your optimization algorithms
- *MULTICUBE Explorer is **not** "yet another simulator" (there already many simulators….)*

Exploration framework: an overview



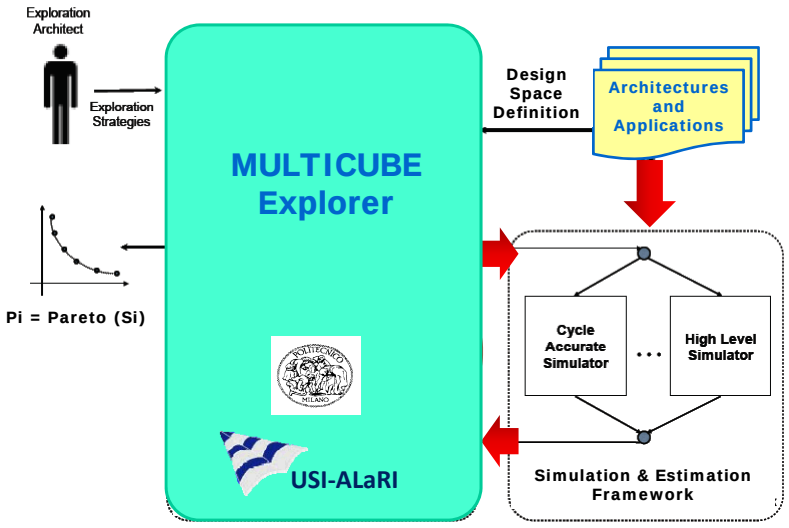
XML-based Tool Interface



Automatic Design Space Exploration: Open Issues

- Efficiency of Automatic DSE process can be improved by:
  1. Minimizing the numbers of simulations to be executed by using **exploration heuristics** such as state-of-art evolutionary algorithms
  2. Speeding up simulations
  3. Simulating at higher abstraction levels
  4. Defining an **analytical response model** of the system behavior based on a subset of simulations to predict the unknown system response

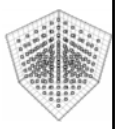
Multi-Objective Design Space Exploration Flow



MULTICUBE Explorer

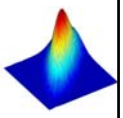
- Multi-Objective DSE framework composed of:

1. Design of Experiments (DoEs):
To identify the experimentation plan: how to select the design points in the design space to be simulated



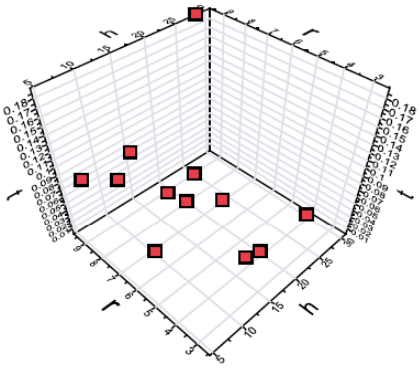
2. Optimisation Algorithms:
Metaheuristics methods inspired by analogies with physics, or with biology to solve multi-objective optimization problems.
This class of methods includes between the others: simulated annealing, genetic algorithms, evolutionary strategies.

3. Response Surface Modeling (RSM):
To use the set of simulated points to obtain a response surface of the system behavior

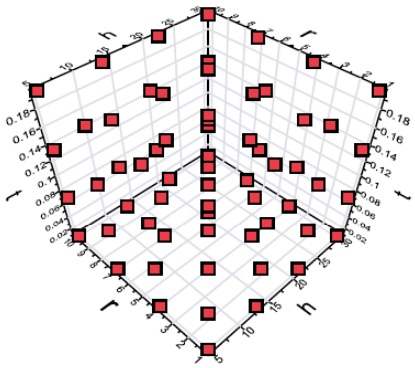


How to explore the design space?

- **Design of Experiments:** to identify the planning of experimentation campaign where the set of tunable design parameters can vary
- To specify the **layout:** how to select the design points in the design space



Random DOE, 12 Entries



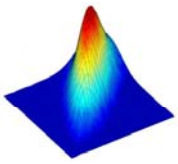
Full Factorial DOE, 64 Entries

Optimisation Algorithms

- Several algorithms can be selected for solving different problems:
  - APRS: Adaptive windows Pareto Random Search
  - MOSA: Multi-Objective Simulated Annealing
  - MOPSO: Multi-Objective Particle Swarm Optimizer
  - NSGA-II: Non-dominated Sorting Genetic Algorithm
  - SEMO: Simple Evolutionary Multi-objective Optimizer
  - FEMO: Fair Evolutionary Multi-objective Optimizer
  - GEMO: Greedy Evolutionary Multi-objective Optimizer
- All these metaheuristics are not mutually exclusive. It is often hard to predict with certainty the efficiency of a method when it is applied to a problem. This statement is confirmed by the well-known *"no-free-lunch theorem"*

Response Surface Modeling

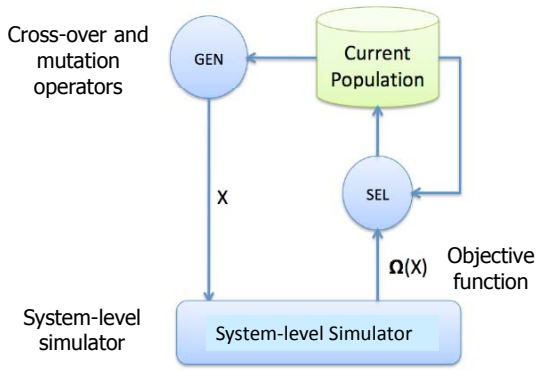
- RSM techniques are used to define an analytical dependence between design parameters and one or more response variables.
- RSM based on two main phases:
  - During the **training phase,** known data (or training set) defined by DoEs are used for tuning the RSM.
  - During the **prediction phase,** the RSM is used to predict the unknown system response.



- Several RSM techniques:
  - Linear Regression
  - Spline Interpolation
  - Shepard's Interpolation
  - Artificial Neural Networks (3-layer fully-connected feed-forward ANNs)
  - Radial Basis Functions
  - Kriging Interpolation (recently added)

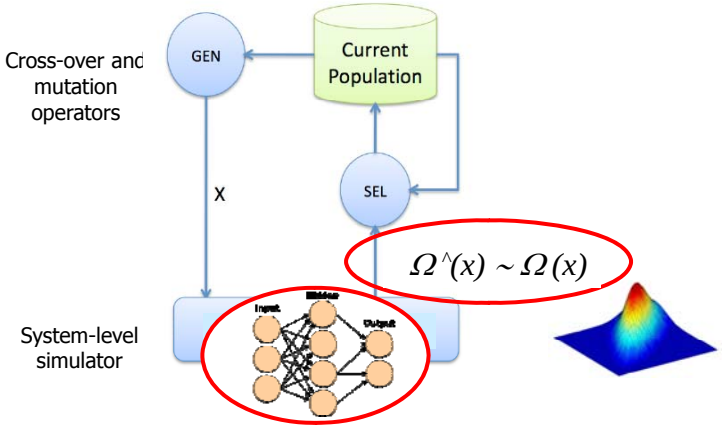
How to combine optimisation heuristics and Response Surface Models?

NSGA-II Exploration Flow



Main problem: Very long simulation time required to evaluate the system-level objective function *Ω(x)*

NSGA-II Exploration Flow combined to ANN model

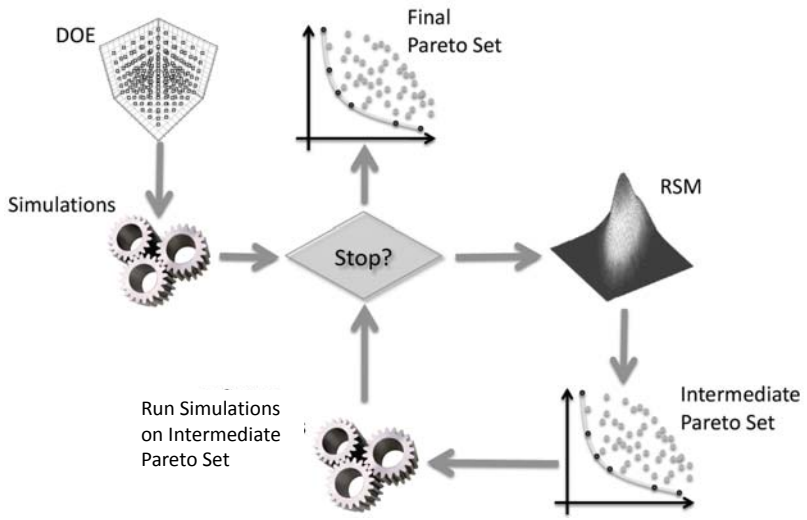


Proposed solution: NSGA-II combined to an Artificial Neural Network to predict the system-level objective function $\Omega(x) = [\varepsilon(x), \delta(x)]$

How to combine DoEs and Response Surface Models?

"ReSPIR: A Response Surface-Based Pareto Iterative Refinement for Application-Specific Design Space Exploration", Palermo, G.; Silvano, C.; Zaccaria, V., *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* Volume 28, Issue 12, Dec. 2009 Page(s):1816 – 1829

ReSPIR: RSM-Support Iterative Pareto Refinement



Target MP-SoC Architecture

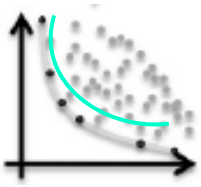
- MIPS-based shared memory MP-SoC with private caches
- Modeled with SESC simulator with power estimation support

Parameter	Min.	Max.
# Processors	2	16
Processor issue width.	1	8
L1 instruction cache size	2K	16K
L1 data cache size	2K	16K
L2 private cache size	32K	256K
L1 instruction cache assoc.	1w	8w
L1 data cache assoc.	1w	8w
L2 private cache assoc.	1w	8w
I/D/L2 block size	16	32

- Design space composed of 2^{17} design points (131 072)
- Four parallel applications {FFT, OCEAN, LU, RADIX} derived from SPLASH-2 benchmark suite for different data-sets

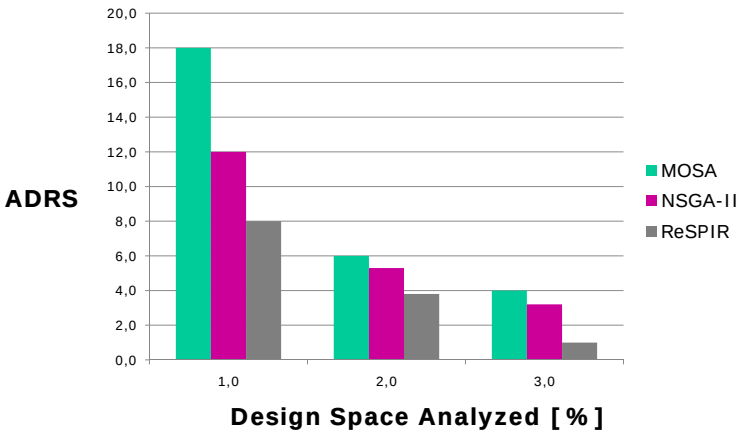
Comparison Results

- Target multi-objective optimization problem: Minimization of ***average execution time*** and ***average [mW per MIPS]***
- The exact Pareto set has been found by exhaustive exploration to be used as reference set
- Accuracy in terms of **Average Distance from Reference Set** to measure the distance between reference Pareto set and approximated Pareto set
  - **Lower ADRS, best approximated Pareto set**



- Comparison with state-of-the-art heuristics:
  - **Multi-Objective Simulated Annealing (MOSA)**
  - **Non-dominated Sorting Genetic Algorithm (NSGA-II)**

Accuracy vs. Design Space Analyzed [%]



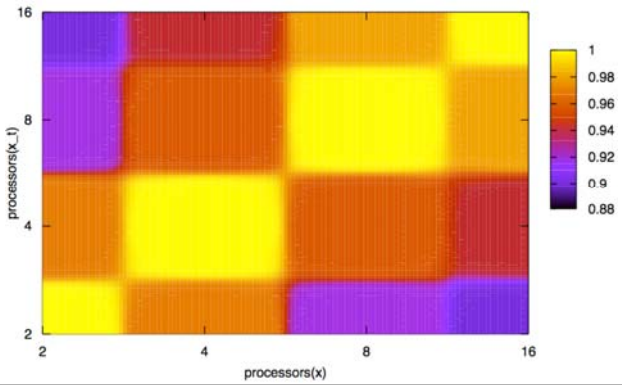
Accuracy in terms of **Average Distance from Reference Set** by varying the percentage of the design space analyzed from 1% to 3%

How to exploit the spatial correlation existing in the design space?

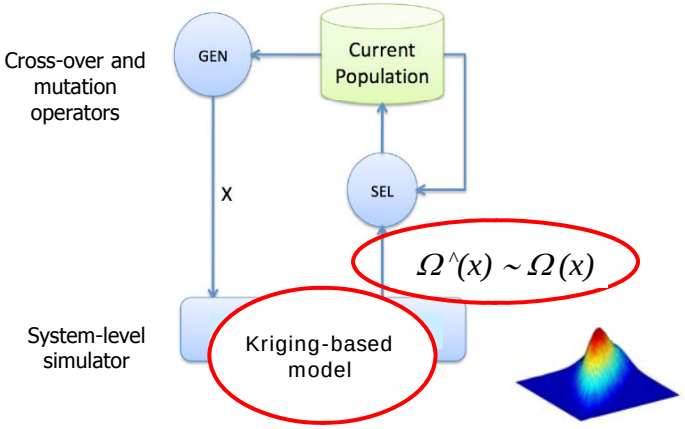
G. Mariani, G. Palermo, V. Zaccaria, A. Brankovic, J. Jovic, C. Silvano. "A Correlation-based Design Space Exploration Methodology for Multi-Processor Systems on-Chip" In Proceedings of **DAC 2010 Design Automation Conference,** Anaheim, CA, USA, June 2010.

Spatial Correlation

- Basic assumption: spatial correlation exists within the design space.
- For the specific use case, correlations between close design points $<x, x_t>$ measured in terms of their performance values $<\delta(x), \delta(x_t)>$
- Basic idea: To exploit the spatial correlation in the design space to build a meta-model of the system behavior



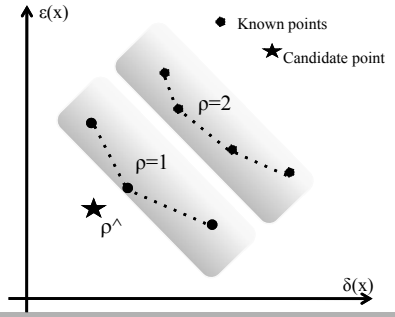
Model Assisted NSGA-II (MA-NSGA-II)



Proposed solution: NSGA-II combined to a Kriging-based model as correlation-based model to predict the system-level objective function $\Omega(x) = [\varepsilon(x), \delta(x)]$

Further step: Metamodel to estimate the non-domination rank

- Reduce the multi-objective problem dimension to minimize the non-domination rank $\rho(x)$ used to represent how much deeply a configuration x is nested into sub-optimal solutions.
  The current Pareto front has non-domination rank $\rho = 1$
- A correlation-based Kringing-based model has been used to predict the non-domination rank $\rho\hat{}(x)$ to identify the best candidate x to be simulated

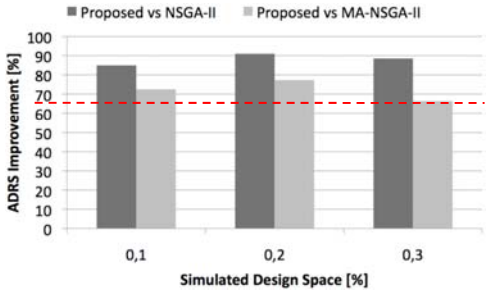


Comparison Results [DAC2010]

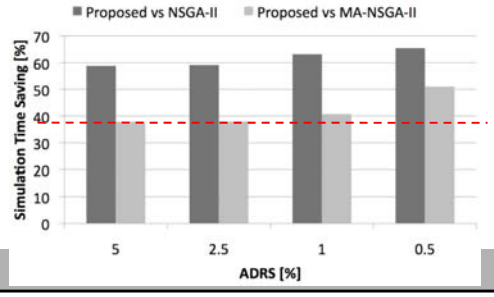
- The exact Pareto set has been found by exhaustive exploration to be used as reference set.
- We compared the accuracy of different state-of-the-art DSE techniques by using the **Average Distance From Reference Set (ADRS).**

- Comparison with 2 reference DSE techniques:
  - Non-domination Sorting Genetic Algorithm **(NSGA-II)**
  - Model Assisted NSGA-II **(MA-NSGA-II)**
    - At each generation the best 40% of individuals are simulated, the others are approximated by the Kriging-based model

Comparison Results [DAC2010]



Quality of the solution set improved by 65%



Exploration time reduced by 37%

Further Step: Run-time Resource Management based on Design-time Exploration

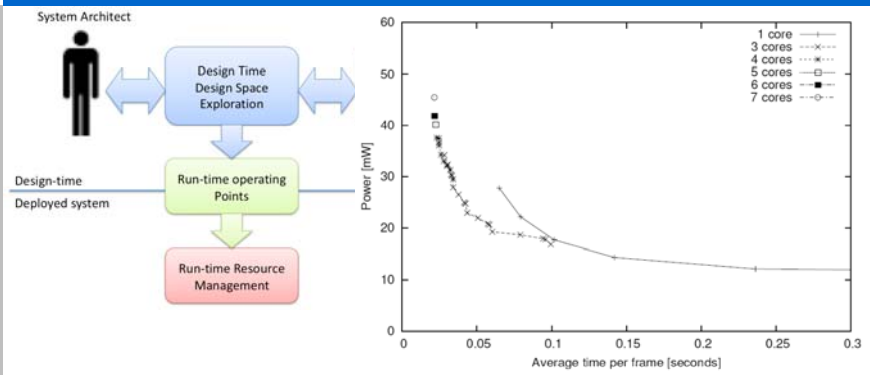


G. Mariani, V. Zaccaria, G. Palermo, P. Avasare, G. Vanmeerbeeck, C. Ykman-Couvreur, C. Silvano, "An industrial design space exploration framework for supporting run-time resource management on multi-core systems ", In Proc. of **DATE 2010 - International Conference on Design, Automation and Test in Europe.** Dresden, Germany. March 2010.

Introduction & Motivations

- Usually several applications running in parallel on the target platform architecture are competing for the access to **system resources**
- User requirements (performance and power) can change **dynamically**
- System configurations providing high performance are power hungry
- We cannot tune at design time the system for peak performance, **design-time optimization is not enough**
- **Run-Time Management (RTM)** of **run-time tunable parameters** (e.g, resource allocation and operating frequencies) is needed to be combined to design-time optimization.

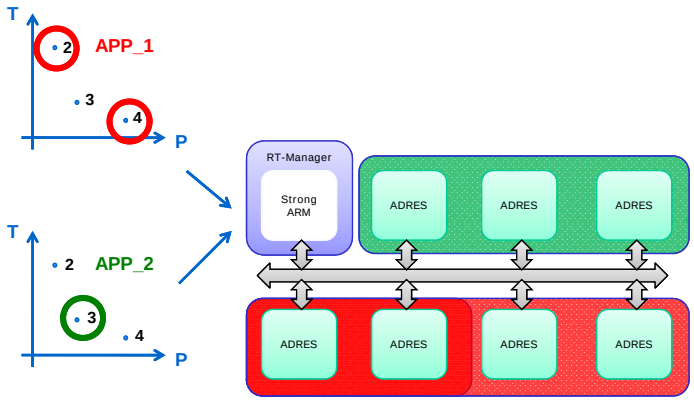
Pareto Optimal Run-Time Operating Points



- Based on the results of **design-time exploration,** we derive a set of Pareto optimal **operating points** corresponding to a power cost, resources (number of cores) and QoS (average time per frame).
- The operating points will be used by the **Run-time Resource Manager** to achieve QoS requirements (average time per frame) while meeting overall resources (number of cores) and minimizing power consumption

Run Time Management of target multi-core platform

- The system state can change due to some events:
  - A new application executed or
  - QoS requirements modified



Conclusions

- An automatic design space exploration methodology has been proposed leveraging Design of Experiments and Response Surface Modeling techniques
- The proposed framework makes automatic exploration of multi-core architectures more feasible
- The proposed design-time exploration has been combined with a run-time resource manager to support run-time decision making
- Future work: Run-time management of applications' parallelism and dynamic compilation support
- This work is part of the ICT-FP7 EU project MULTICUBE

www.multicube.eu

MULTICUBE Project

MULTI-OBJECTIVE DESIGN SPACE EXPLORATION OF MULTI-PROCESSOR SOC ARCHITECTURES FOR EMBEDDED MULTIMEDIA APPLICATIONS

www.multicube.eu

Project Duration: from January 2008 to June 2010

Politecnico di Milano (POLIMI) – Italy (Project Coordinator)

DS2 – Spain

IMEC - Belgium

STMicroelectronics - Italy

ESTECO - Italy

Università della Svizzera Italiana (ALaRI) - CH

University of Cantabria - Spain

STMicroelectronics - China

Institute of Computing Technology (ICT) China