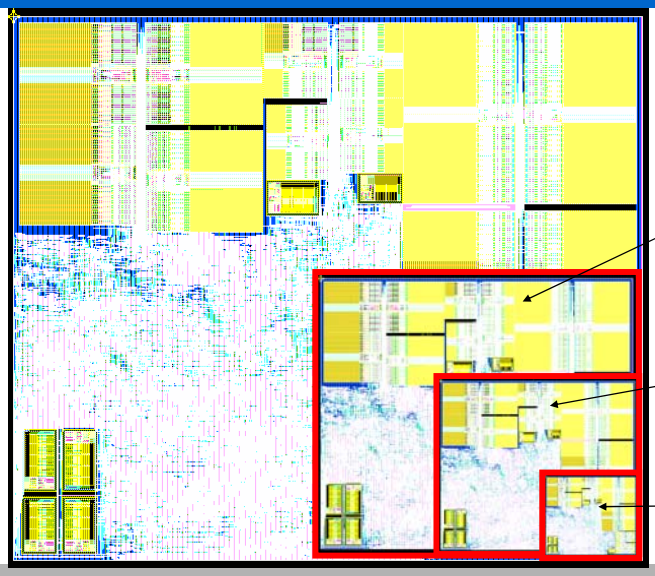
Automatic Design Space Exploration for Chip-Multi Processors

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

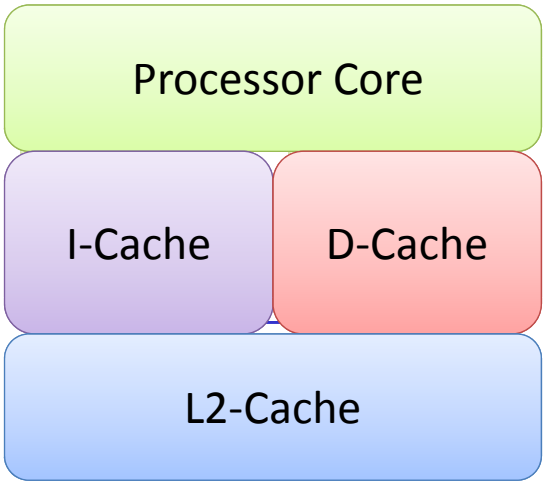
Introduction and Motivations

Squeezing of computing cores



Source: STMicroelectronics

The age of multi / many-core architectures



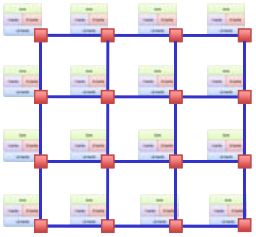
The design space

- 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**

5

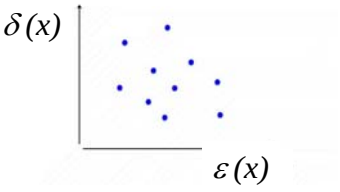
The multi-objective optimisation problem

- 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



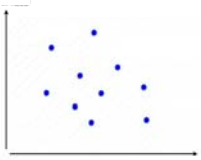
6

The concepts behind the automatic DSE

Output Variables define the objective space

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

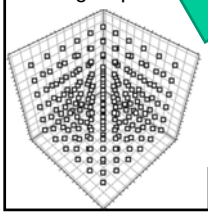
$\delta(x)$



$\varepsilon(x)$

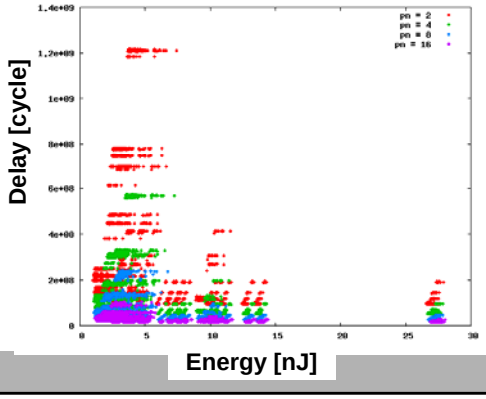
The black box can be:
1) A simulator that models the system behavior and generates output values
2) A set of solvers that models the system behavior and estimates output values

Input Variables: architecture parameters that define the design space



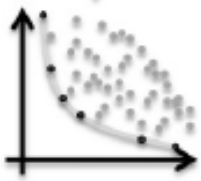
Full Search Design Space Exploration

- 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



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)**



9

Motivations

- Why Automatic Design Space Exploration?

  1. Faster exploration time
  2. Better quality of results

10

Automatic Design Space Exploration: The MULTICUBE Approach

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** focused 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 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

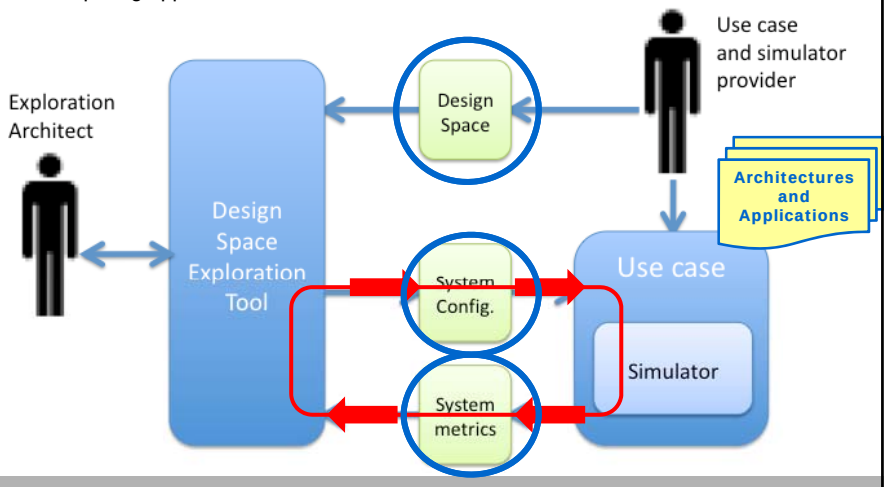
University of Cantabria - Spain

STMicroelectronics - China

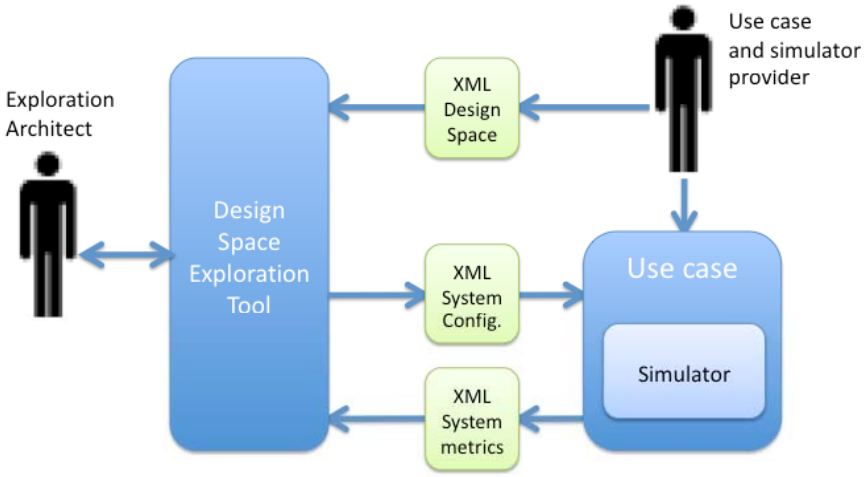
Institute of Computing Technology (ICT) China

Automatic Design Space Exploration Flow

- The **MULTICUBE design flow** for automatic Design Space Exploration of complex architectures such as Multiprocessor Systems on Chip designed for embedded computing applications

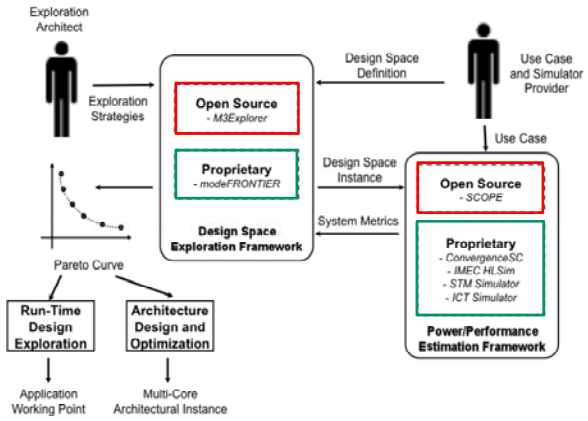


XML-based Tool Interface

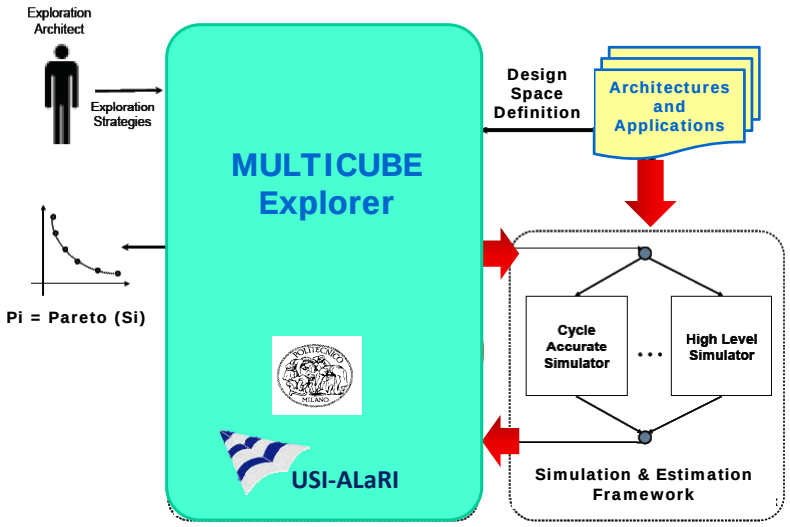


Automatic Design Space Exploration Flow (cnt'd)

- The design flow is based on the co-existence of **open source** and **proprietary** tools based on a **common standard interface.**



Multi-Objective Design Space Exploration Flow

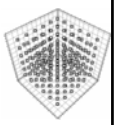


17

MULTICUBE Explorer

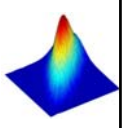
- Multi-Objective DSE framework composed of:

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.

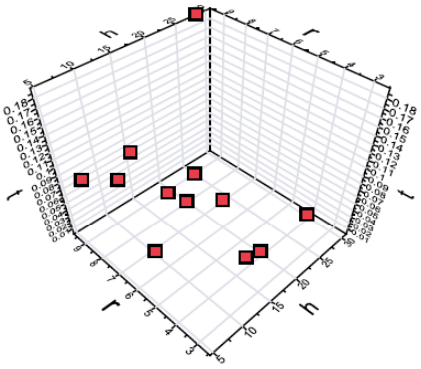
3. Response Surface Modeling (RSM):
To use the set of simulated points to obtain a response surface of the system behavior



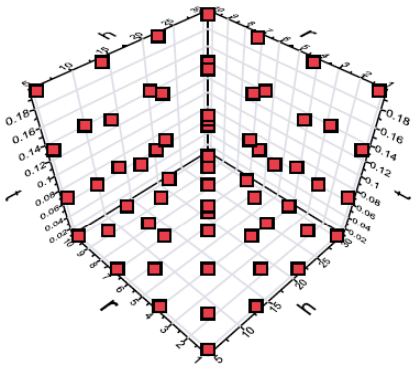
18

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

19

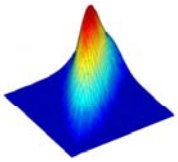
Optimisation Algorithms

- 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"*

20

Response Surface Modeling

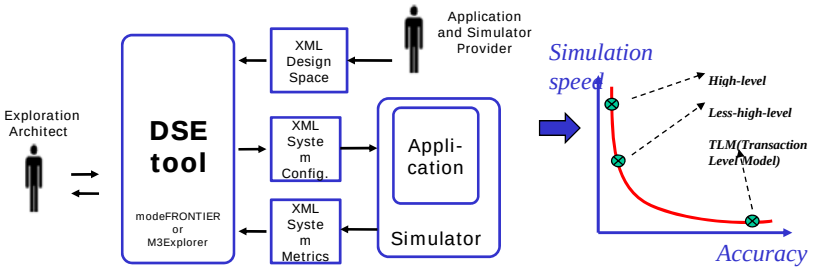
- RSM techniques are used to define an analytical dependence between design parameters and one or more response variables.
- RSM based on two main phases:
  - During the **training phase,** known data (or training set) defined by DoEs are used for tuning the RSM.
  - During the **prediction phase,**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

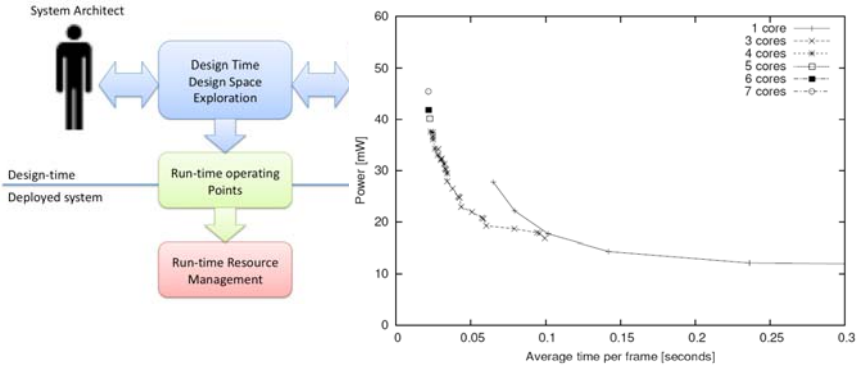
21

A Multi-Simulator based DSE Approach



- The proposed methodology based on the synergy of multiple simulators at several abstraction level can be exploited to further speed up the DSE process while guaranteeing the accuracy of the simulation results.
- The methodology has been validated for the MPEG4 encoder application provided by IMEC by using three different simulators (M3SCoPE, HLSim and TLM-based simulator) interfaced with the two available DSE tools (modeFRONTIER and M3Explorer).
- The Multi-Simulator based DSE Approach is fully described in the **public** deliverable **D4.2.4** available at the MULTICUBE web site

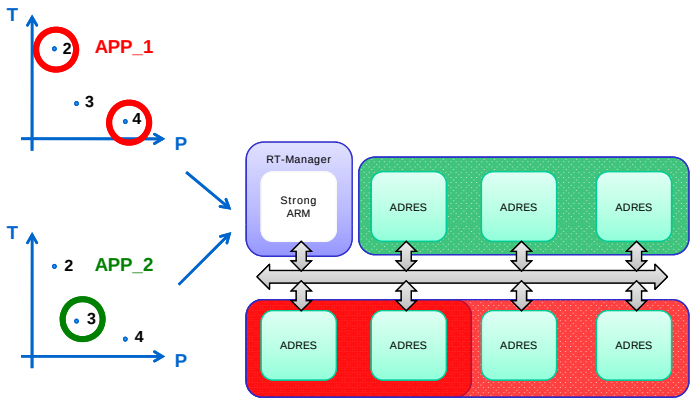
Pareto Optimal Run-Time Operating Points



- 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



Validation and Assessment

Four Use Cases and Demonstrators

The use cases (defined as combination of application and related architecture) have been selected as demonstrators to validate the Integrated Design Tool Prototype in the following contexts:

- **DS2 Powerline communication:** an advanced telecommunication gateway to transfer audio, video and data streaming through the domestic power line

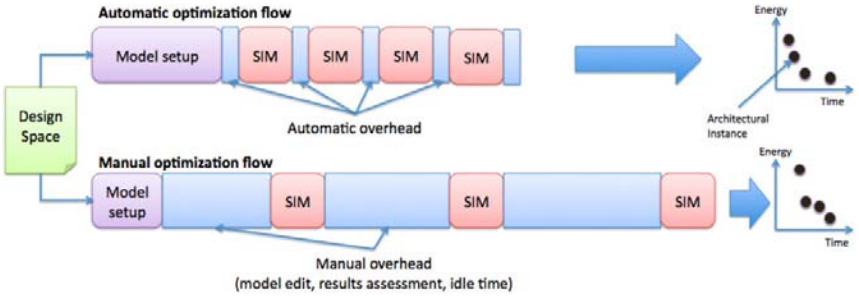
- **Multimedia:** a multiprocessor virtual platform for high data rate processing and compression of high-definition video (MPEG4) (Public Demonstrator)

- **Advanced computing platform:**
  - **ICT Multi-core architecture: a tiled homogeneous multi-core architecture for general embedded purpose (Godson-T)**

  - **STMicroelectronics ultra low power processor: a 4-issue superscalar processor for embedded mobile (Loongson™ - STLS)**

Automatic DSE Assessment Procedure

- Definition of a procedure to assess the benefits of the introduction of Automatic Design Space Exploration in the design process



Testimonials

DS2: *"The proposed flow can save up to 80% of designer time while achieving better results in terms of performance since much more simulations can be run and analysed following this flow obtaining ten times more possible combinations than with the semi-automatic design flow".*

STM: *"The use of MULTICUBE optimisation flow can save up to a 73% of the overall time and achieving comparable results in terms of power/performance tradeoffs.... The overall conclusion of the utilization of MULTICUBE design flow over a real industrial use case like the one just discussed is that the added value overcomes largely the cost of setting up the whole process, converging much faster to the optimum solution for a given technical problem."*

28

Testimonials (cont'd)

- **IMEC:** *"Looking at the two design space exploration case-studies, it can be seen that using automatic DSE over manual full-space exploration has large benefits in terms of time-to market and accuracy of exploration results. By using automatic DSE in dynamic runtime management evaluation case-study, design space exploration time was reduced from 6 months (full-space) to 36 hours. In another case-study of MPSoC platform optimisation, by using a simulator at higher abstraction level coupled with automatic DSE, we could reduce design space exploration time from worst case of 153 years (full-space) to 36 hours… The extra efforts required to build automatic DSE procedure were minimal".*

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
- This work was part of the ICT-FP7 EU project MULTICUBE

www.multicube.eu