



A Design Space Exploration Framework for Run-Time Resource Management on Multi-Core Architectures

Cristina Silvano

Politecnico di Milano, Milano (ITALY)
Dipartimento di Elettronica e Informazione
silvano@elet.polimi.it
<http://home.dei.polimi.it/silvano/>



Outline

- Introduction and Motivations
- MULTICUBE Explorer: an Automatic Design Space Exploration Framework
- Run-time Resource Management
- Case Study
- Conclusions



Introduction and Motivations



Introduction and Motivation

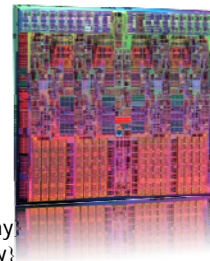
- Given the increasing **complexity** of Chip Multi-Processors, a wide range of architecture parameters (number of processors, processor issue width, L1 & L2 cache sizes, etc.) must be tuned

- Design space of the target architecture A should consider all possible configurations of each parameters p_i :

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

- Example:

- Number of Processors = {2, 4, 8, 16}
- Processor Issue Width = {1, 2, 4, 8}
- L1 Instr. Cache Size = {2KB, 4KB, 8KB, 16KB}
- L1 Data Cache Size = {2KB, 4KB, 8KB, 16KB}
- L2 Private Cache Size = {32KB, 64KB, 128KB, 256KB}
- L1 Instr. Cache Associativity = {1-way, 2-way, 4-way, 8-way}
- L1 Data Cache Associativity = {1-way, 2-way, 4-way, 8-way}
- L2 Data Cache Associativity = {1-way, 2-way, 4-way, 8-way}
- I/D/L2 Cache Block Size = {16, 32}



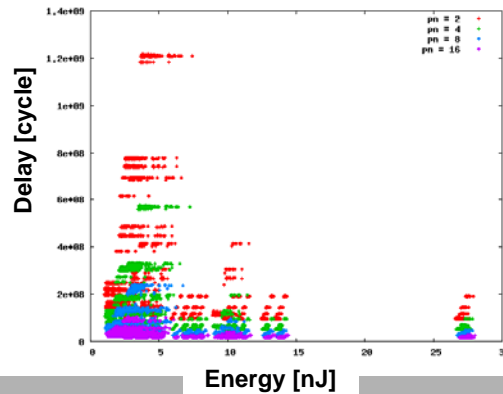
⇒ Huge design space composed of 2^{17} (131 072) system configurations



Full Search Design Space Exploration



- In most cases, the design space to be explored is **huge**
- Automatic Design Space Exploration based on **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 1 min \Rightarrow 131 072 min \sim 3 months for full-search exploration



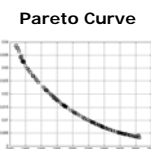
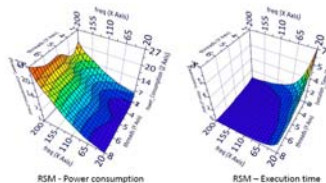
7



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 based on intuition and past experience of the design architects
- **Need for Automatic Design Space Exploration** to support systematically the exploration and the quantitative comparison in terms of multiple competing objectives (**Pareto analysis**)

MULTI-OBJECTIVE EXPLORATION



8

Cristina SILVANO - Politecnico di Milano (ITALY)



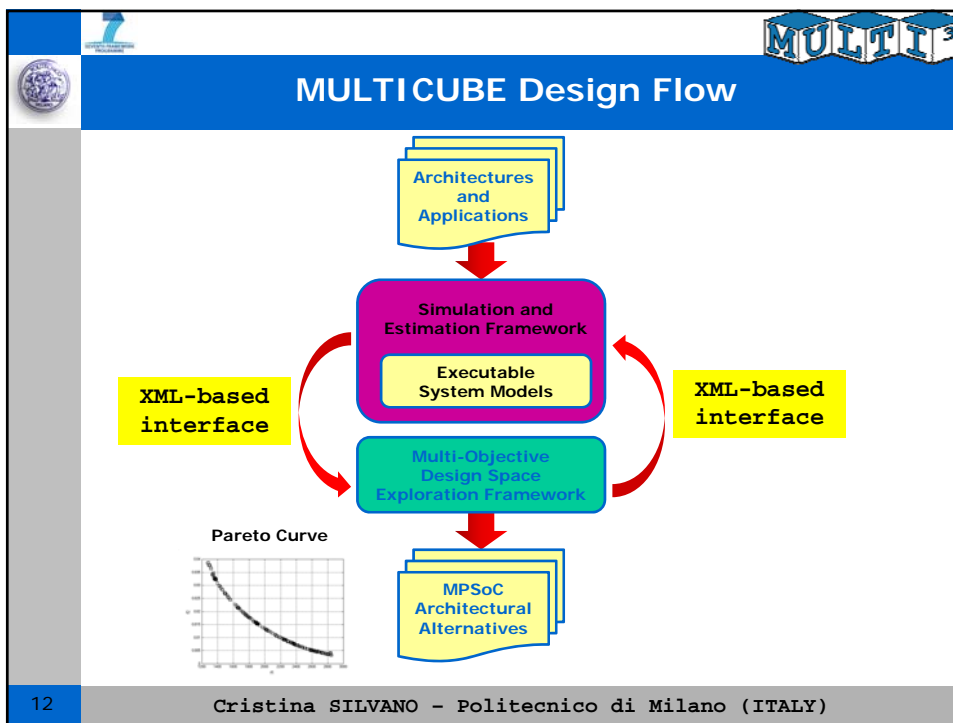
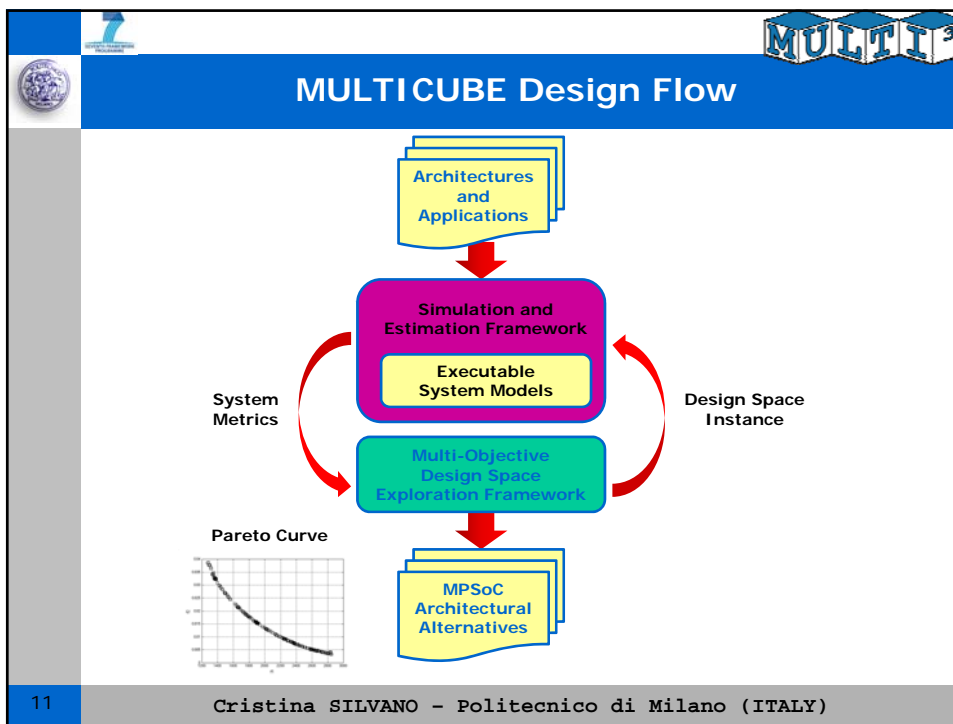
Automatic Design Space Exploration



MULTICUBE Project



- So far, several heuristic techniques have been proposed to address the design space exploration problem, but they are all characterized by low efficiency
- An **overall design space exploration framework** is needed to combine all optimizations into a global search space with a common interface to the simulation and evaluation 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.

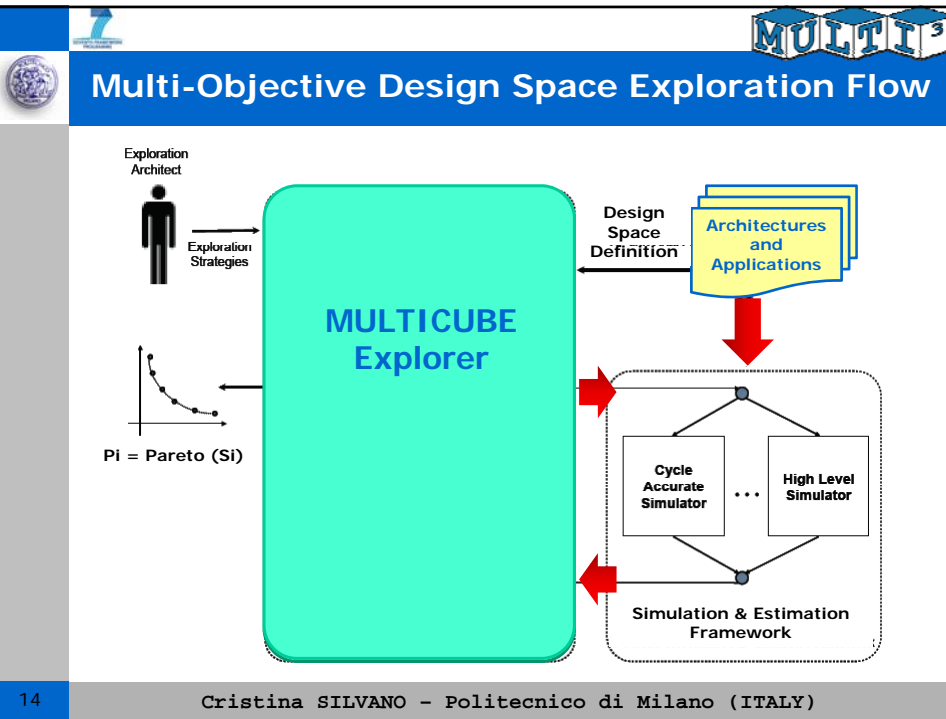




Automatic Design Space Exploration

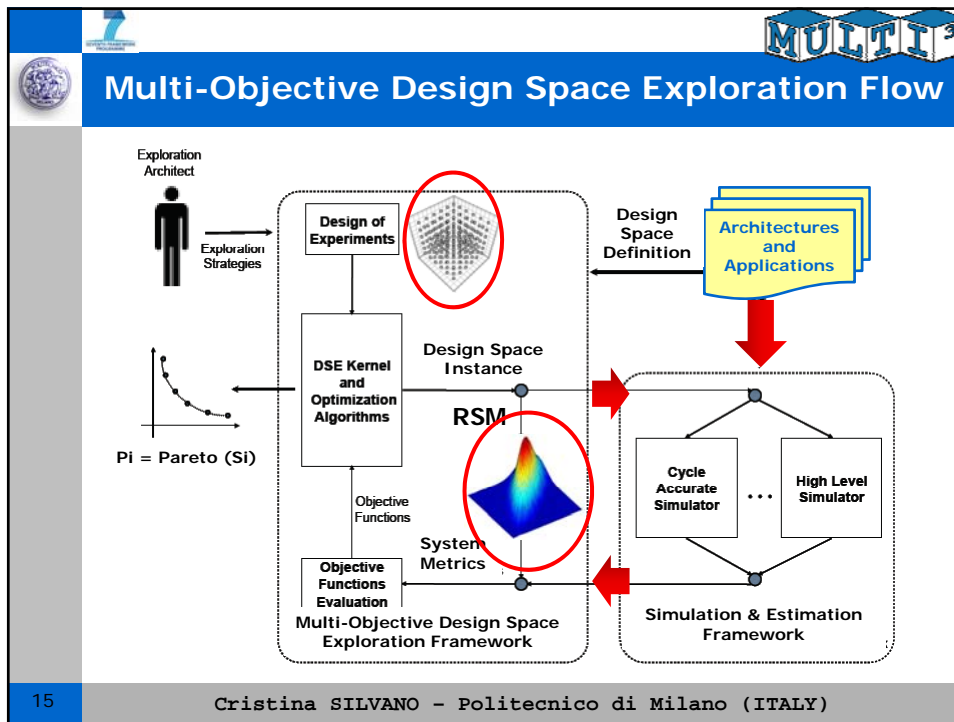
- Efficiency of DSE process can be improved by:
 1. Minimizing the numbers of simulations to be executed by using **exploration heuristics** such as state-of-art evolutionary algorithms
 2. Speeding up simulations
 3. Simulating at higher abstraction levels
 4. Defining an **analytical response model**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

13



14

Cristina SILVANO - Politecnico di Milano (ITALY)



15



Cristina SILVANO - Politecnico di Milano (ITALY)

Multi-Objective Design Space Exploration

- Multi-Objective DSE framework based on:
 - Design of Experiments (DoEs):** To identify the experimentation plan where the set of tunable design parameters can vary
 - Response Surface Modeling (RSM):** To use the set of data generated by DoE to obtain a response surface of the system behavior
- RSM based on two main phases:
 - During the **training phase**, known data (or training set) are used for tuning the RSM.
 - During the **prediction phase**, the RSM is used to predict the unknown system response.

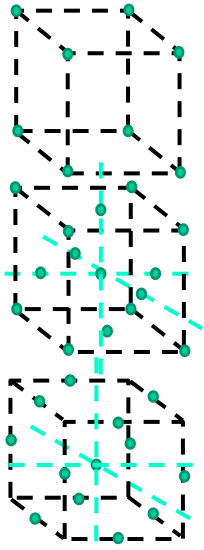
16

Cristina SILVANO - Politecnico di Milano (ITALY)






Design of Experiments

- Identifies the planning of experimentation campaign where the set of tunable design parameters can vary
- Specifies the **layout**: how to select the design points in the design space
- **Four DoE techniques have been applied:**
 - **Random**
 - **Full Factorial**
 - **Central Composite**
 - **Box Behnken**

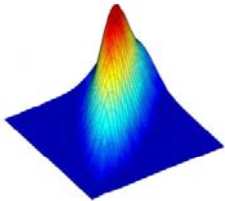


17
Cristina SILVANO - Politecnico di Milano (ITALY)

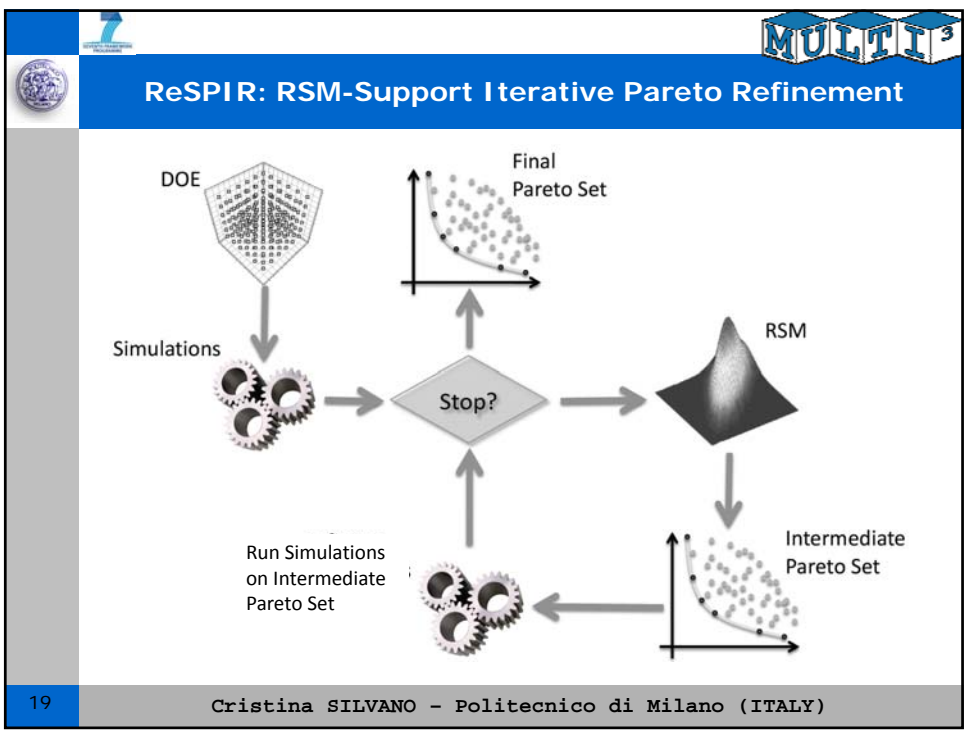



Response Surface Modeling

- RSM techniques are used to define an analytical dependence between design parameters and one or more response variables.
- To use the set of data generated by DoE to obtain a response model of the system behavior to forecast unknown system response.
- Four techniques have been applied:
 - RSM based on Linear Regression
 - RSM based on Shepard's Interpolation
 - RSM based on Artificial Neural Networks
 - Three-layer fully-connected feed-forward ANNs
 - RSM based on Radial Basis Functions



18
Cristina SILVANO - Politecnico di Milano (ITALY)



19

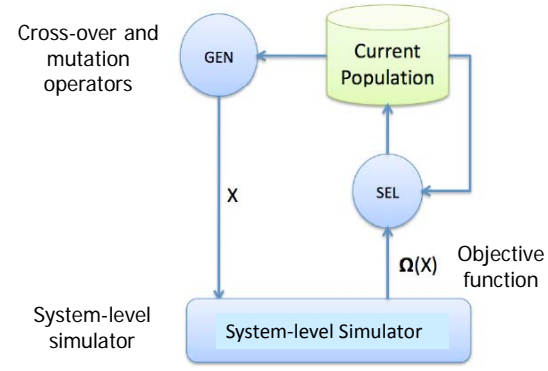
Cristina SILVANO - Politecnico di Milano (ITALY)



ANN-model assisted NSGA-II Exploration Flow



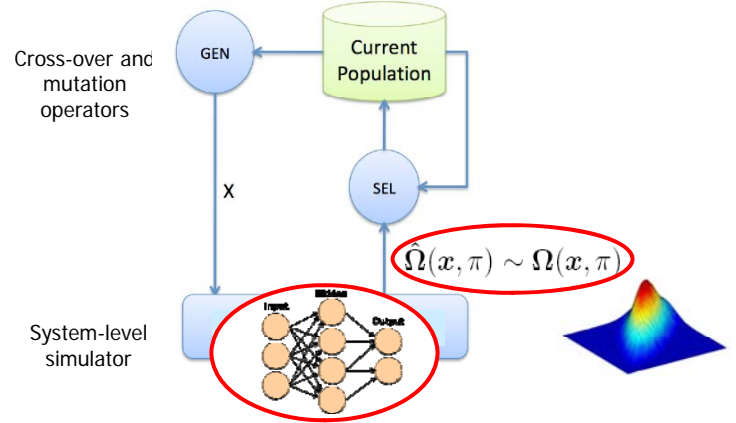
NSGA-II Exploration Flow



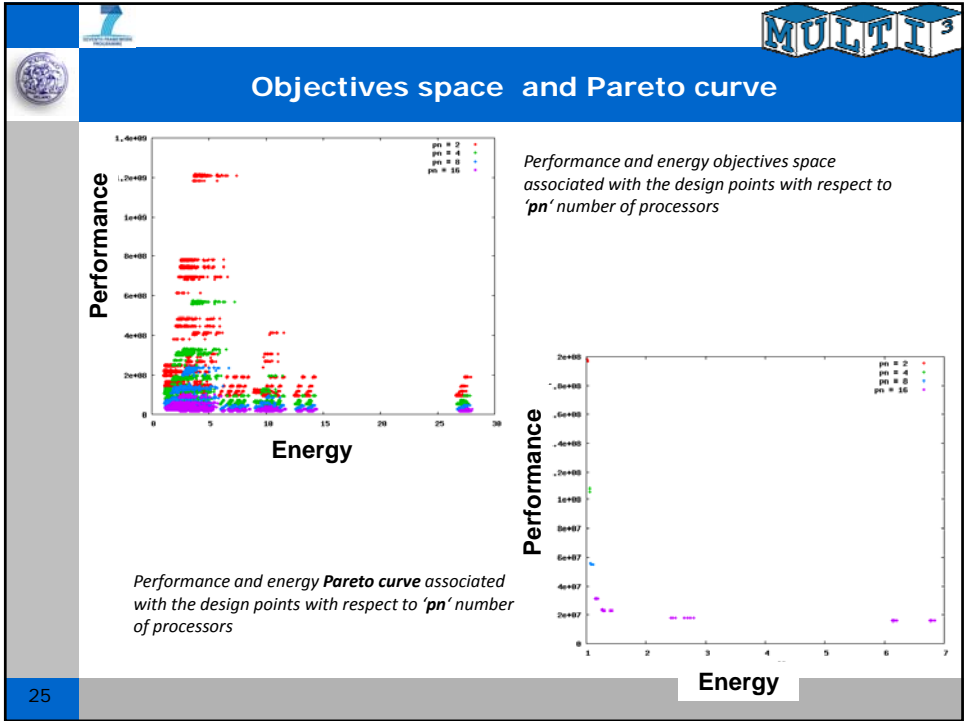
Main problem: Very long simulation time required to evaluate the system-level objective function $\Omega(x, \pi)$



ANN-model assisted NSGA-II Exploration Flow



Proposed solution: NSGA-II assisted by an Artificial Neural Network to predict the system-level objective function




Run-time Resource Management





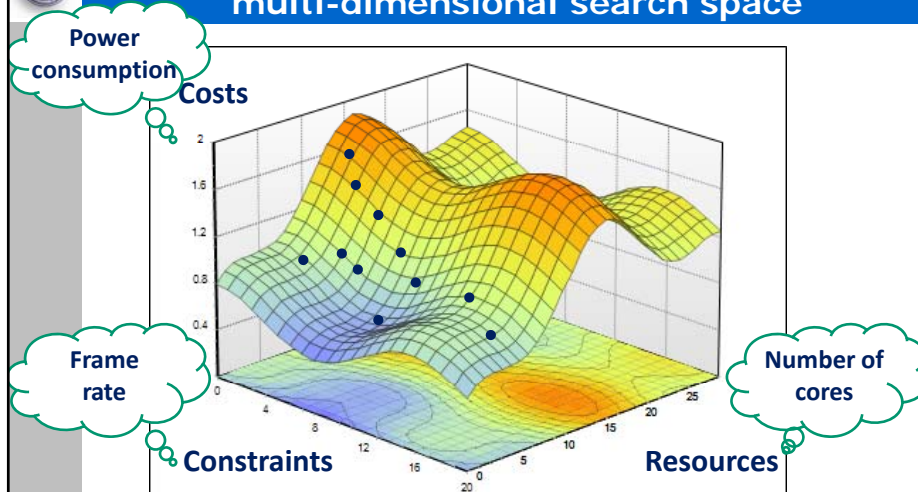


Introduction & Motivations

- Usually several **applications** running in parallel compete 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 with design-time optimization.



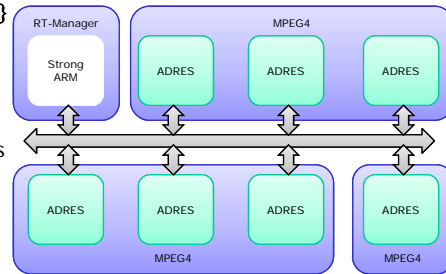
Application operating points in the multi-dimensional search space





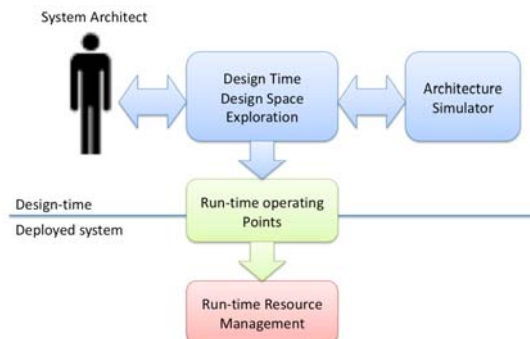
Target Multi-core Architecture

- **8-Core IMEC Platform Architecture:**
 - 1 Strong ARM core to run the run-time manager
 - 7 ADRES VLIW cores with dynamic frequency scaling
- **Application:**
 - Automotive Cognitive Safety System with 3 MPEG4 encoder applications
 - Frame Rate is considered as QoS (i.e, the user requirement)
- **Tunable parameters:**
 - ADRES-core frequency
 $\Phi = \{20, 60, 100, 140, 180, 220\}$
 - Number of ADRES cores $\{1, 3, 4, 5, 6, 7\}$ allocated to each MPEG4 application
 - Task-level parallelism of each MPEG4 application on multiple cores can be changed by *code versioning*



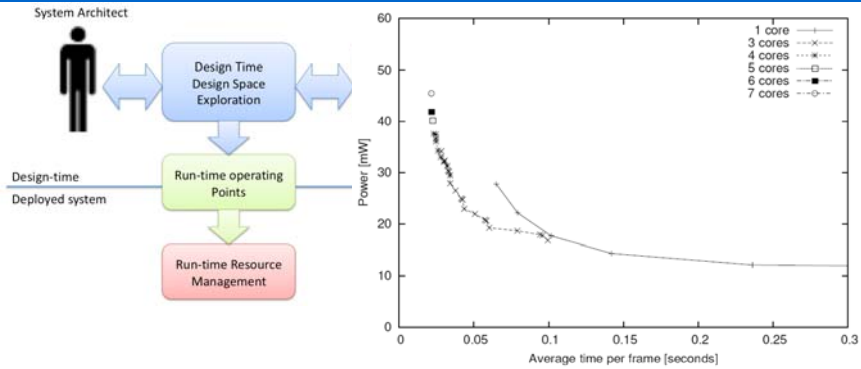
Design-time Exploration & Run-time Manager

- For each application, **Design Time Exploration** of run-time configurable parameters
 - **Result:** Set of Pareto-optimal run-time operating points
- For each active application, a **Run-time Manager** has to select at run-time the working points from the set of Pareto-optimal operating points





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, used 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 (up to 7 ADRES cores) and minimizing power.



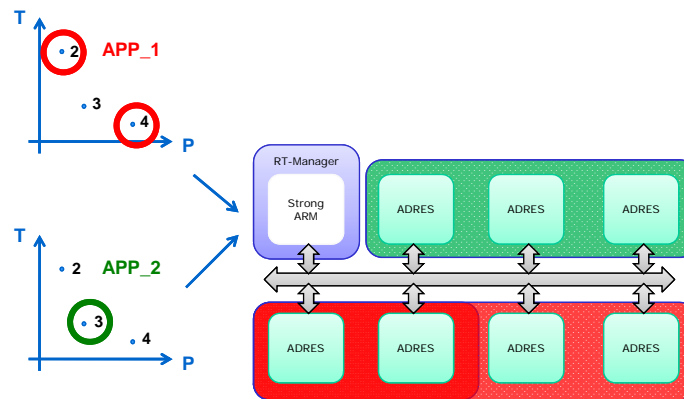
Run-time Resource Management Problem

- **Given:**
 - A set of active applications $\alpha \in A$ ranked by a priority $\omega(\alpha)$
 - A set of Pareto operating points C_α for each α derived from design-time exploration
 - A set of events changing QoS constraints τ_{max}^α for each $\alpha \in A$
- **Problem: To assign at run-time the operating point c_α to each active application:**
 - Meeting resource constraints: $\sum_{\alpha \in A} \rho(c_\alpha) < \text{available resources}$
 - Meeting QoS constraints: $\tau_\alpha < \tau_{max}^\alpha$
 - Minimizing the power consumption: $\sum_{\alpha \in A} \pi(c_\alpha)$
- Multi-dimension multiple-choice Knapsack Problem (MMKP)



RTM Overview

- Every time system state changes due to some events:
 - A new application executed or
 - QoS requirements modified



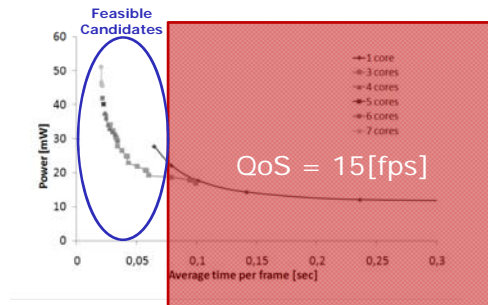
Run-time Resource Manager

- Multi-dimension multiple-choice Knapsack Problem (MMKP)
- RTM: Low-complexity greedy prioritized heuristic to be used at run-time with low computational overhead
- RTM Complexity: $O(\rho N \log(\rho N))$ where N is the average number of operating points per application



Run-time Preprocessing

- Application is activated (e.g. MPEG4)



- The QoS requirement is set (or modified)
- For each candidate we compute an user value as the power saved from the candidate when compared to the most power hungry alternative



Run-time Decision Making Mechanism

- Sort all feasible candidates
- Generate an initial solution

Feasible Candidates				
App	id	U val	Res	Frequencies
1	3	3.5	3	<60,100,60>
3	1	2.1	4	<60,20,20,60>
1	9	2.0	1	<220>
2	4	1.2	1	<180>
3	2	0.3	3	<100,100,100>

Sort by

Lowest resource cost

		Solution		
App Id	Resource Cost	1	2	3
	User value	2.0	1.2	0.3



Run-time Decision Making Mechanism

Optimization:

- Place highest value object in the solution
- Solution is feasible? Save the best solution
- Place next object ...

Feasible Candidates

App	id	U val	Res	Frequencies
1	3	3.5	3	<60,100,60>
3	1	2.1	4	<60,20,20,60>
1	9	2.0	1	<220>
2	4	1.2	1	<180>
3	2	0.3	3	<100,100,100>

Solution

App	1	2	3
Id	3	4	1
Resource Cost	3	1	4
User value	3.5	1.2	2.1

NO FEASIBLE SOLUTION?

- Relax QoS of the application with lowest priority
- Update the Candidate list
- ...

Sort by Lowest resource cost



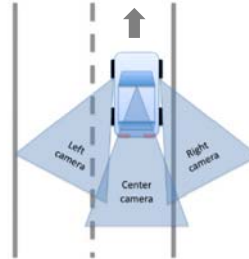
Experimental Results



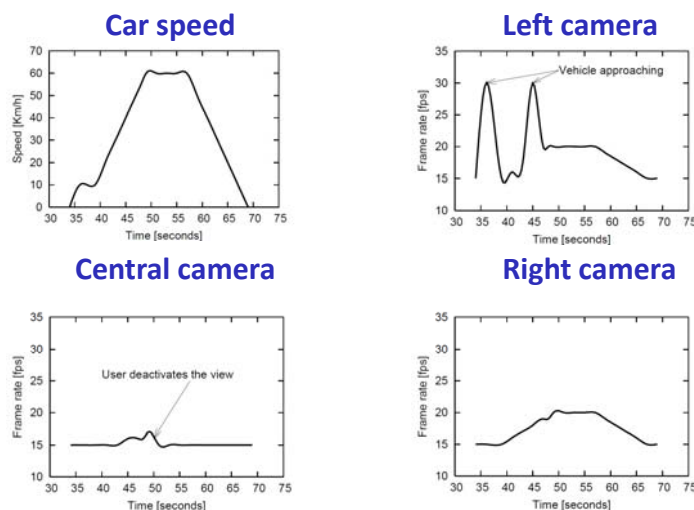


Automotive Cognitive Safety System

- Vehicle with 3 on-board cameras associated with left, center and right mirrors with one MPEG4 encoder per mirror
- Keep passenger safer:
 - Forward collision warning
 - Automatic pre-crash emergency breaking
 - Lane departure warning
 - Lane change assist/blind spot assist
- The driver can switch on/off mirror views on a multiple display dashboard to reproduce the live-streams:
 - Mirrors frame rates depend on car speed and vehicle proximity
 - Priorities:
 - Default: higher priority to central and left mirror
 - If a vehicle is in the proximity, higher priority to lateral mirrors

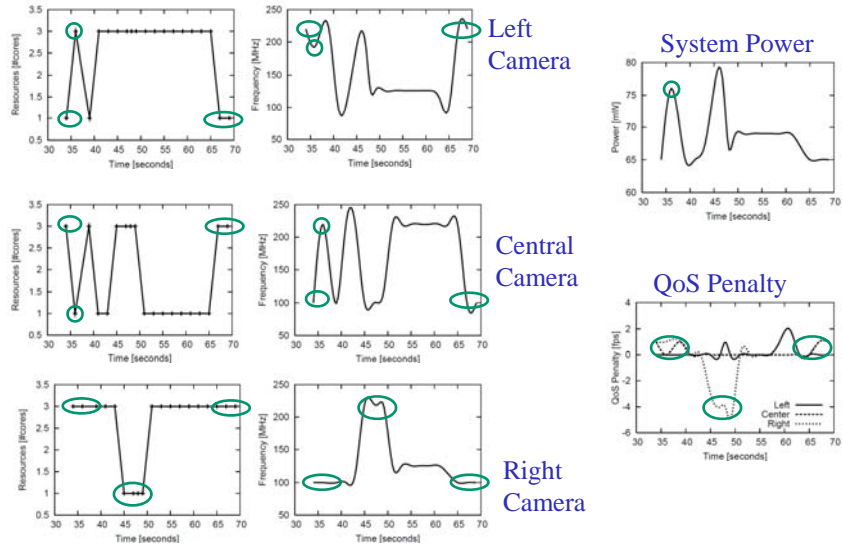


Car speed and QoS frame rate for the 3 on-board cameras (left, center and right)



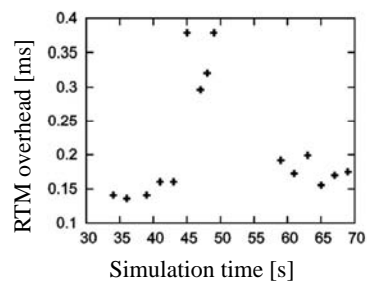




Experimental Results



Execution Time

- Run-Time resource Management (RTM) implemented in C
- RTM simulated on the Strong ARM processor
- RTM overhead at 206 MHz:



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

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

[2] 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,

43 Cristina SILVANO - Politecnico di Milano (ITALY)













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

44 Cristina SILVANO - Politecnico di Milano (ITALY)