Sensitivity Analysis
&
System Robustness Maximization

Short Overview

Arne Hamann
Razvan Racu
Rolf Ernst

Bologna, 22.05.2006

Sensitivity Analysis

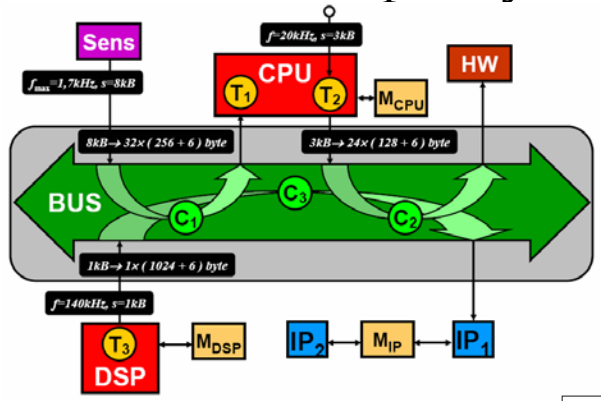
- SA determines “performance reserve” (slack) before a system fails to meet timing constraints
- Helps to identify critical components requiring special focus during design
- Previous approaches consider system sensitivity with respect to variations of a single design property
  - WCETs, input data rates, resource speeds
- Adequate for independent design properties
- In reality components often have complex timing dependencies

→ Can be captured by multi-dimensional SA

Multi-Dimensional Sensitivity Analysis

- Two approaches (ECRTS 06)
  - Search based approach with smart step
    - Based on binary search (base parameter, target parameter)
    - Currently 2-dimensions
    - Smart step: exploits monotonic behavior of the sensitivity front (e.g. for WCET sensitivity)
  - Stochastic approach
    - Based on multi-objective evolutionary search techniques (PISA/SPEA2)
    - Search space: system property modification (i.e. WCETs, CPU clock rates, input data rates, etc.)
    - Optimization objectives: minimize/maximize these system properties
    - Pareto-front corresponds to sensitivity front

Example System



Priorities:
BUS: C3 > C2 > C1
CPU: T1>T2

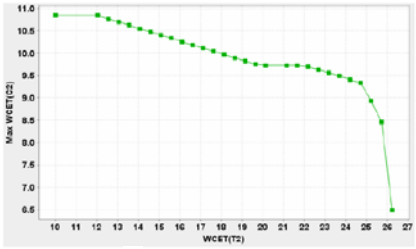
task	execution time	channel	communication time
T_1	250	C_1	17.50
T_2	10	C_2	6.50
T_3	[2;4]	C_3	2.15

Execution and communication times

functional path	max delay
$Sens - C_1 - T_1$	$400\mu s$
$T_2 - C_2 - HW$	$300\mu s$
$T_3 - C_3 - IP_1$	$10\mu s$

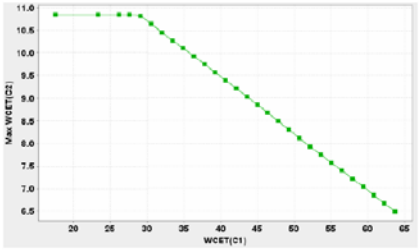
Real-time constraints

Sensitivity Results (2-dimensional)



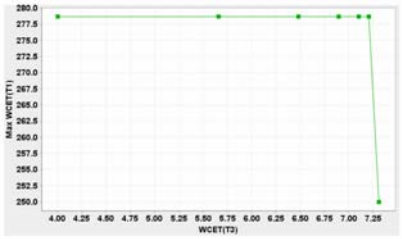
a) $WCET_{T_2} - WCET_{C_2}$

Tasks with functional dependencies



b) $WCET_{C_1} - WCET_{C_2}$

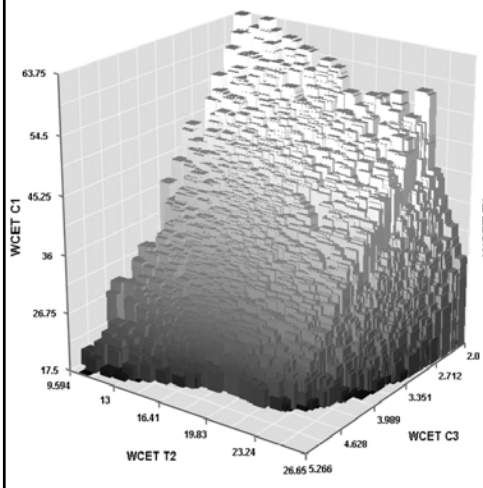
Tasks with load dependencies



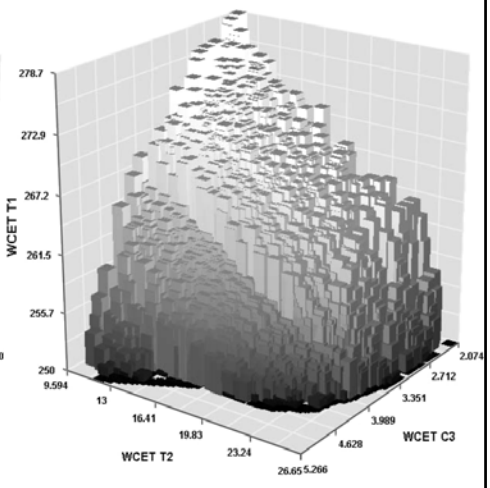
c) $WCET_{T_3} - WCET_{T_1}$

Tasks with no direct dependency

Sensitivity Results (3-dimensional)

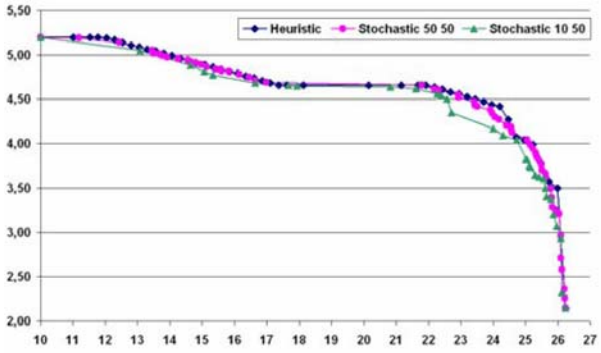


(a) $WCET_{C3} - WCET_{T2} - WCET_{C1}$



(b) $WCET_{C3} - WCET_{T2} - WCET_{T1}$

Accuracy and Complexity



Comparison of the two approaches

search-based			stochastic	
depth	$w/\ s_step$	$w/o\ s_step$	(#gen;#ind)	$WCET$
3	1469	1526	(10;50)	16806
5	4334	4580	(30;50)	50327
7	15906	17924	(50;50)	85141

The run-times of the algorithms (ms)

System Robustness Optimization

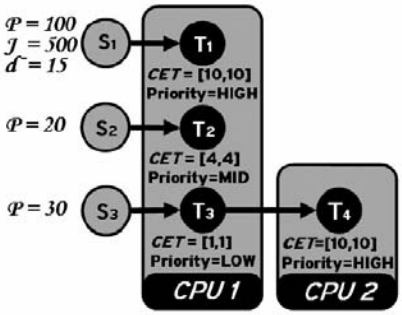
- design properties are subject to modifications
  - during the design process: specification changes, performance estimate changes, exchange of platform components, etc.
  - in the product lifecycle: product updates (HW, firmware, and SW), integration of new components, etc.
- such changes introduce uncertainties and increase design risk
- find approaches to analyze and reduce risk
- Goal: early choose balanced system configuration offering large robustness for critical components

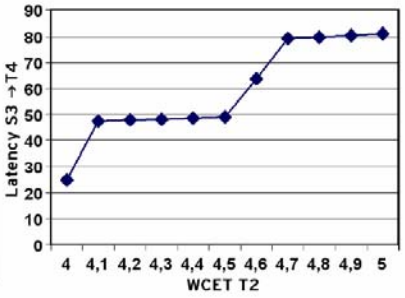
  → Increases system stability and maintainability

Design Property Variations

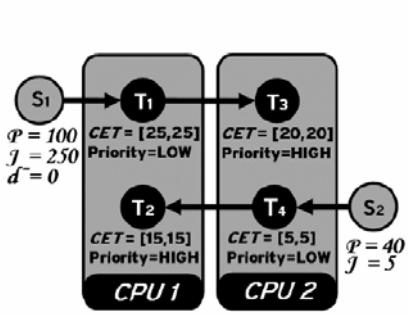
- We consider
  1. Variations influencing the system load
     - Changes of software execution path length and communication volumes
     - Changes of input data rates
  2. Variations influencing the system service capacity
     - Processor and communication link performance changes

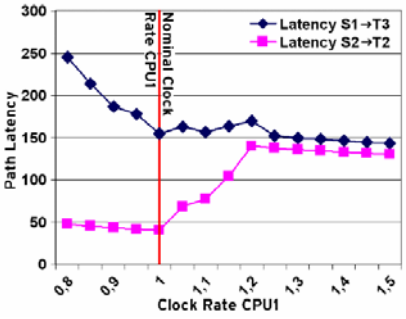
Example 1: WCET Variation





Example 2: CPU performance variation





Robustness definition

- Intuitive definition
  - a system is robust that provides required functionality and meets **contraints** under **system property modifications**
- We introduce robustness metrics based on the notion of slack
  - Given:
    - constrained system *S*
    - parameter configuration *c*
    - System property $p \in S$
  - We define:

$$\text{slack}_{p;c} = \frac{v_c^+(p) - v(p)}{v(p)} \times 100$$

  - were *v(p)* is the current value of p and $v_c^+(p)$ is the maximum property value for p not leading to constraint violations

Static Design Robustness (1)

- Expresses the robustness of a fixed parameter configuration with respect to a set of critical design properties
- design scenario:
  - parameters are defined and fixed early at design time
  - parameters are not modified later to reach compatibility for variants, updates, and bug-fixes
  - state of the practice

Static Design Robustness (2)

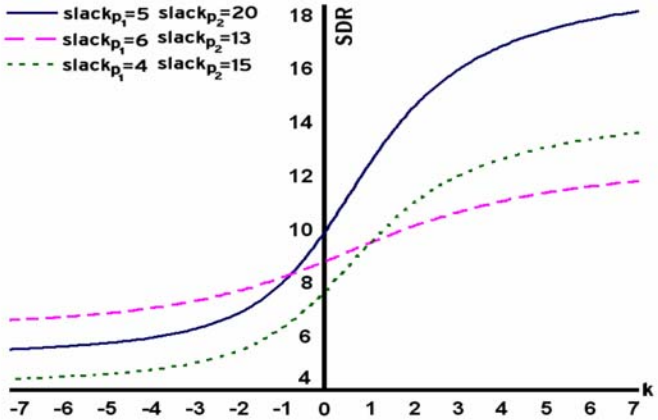
- Given:
  - constrained system S
  - parameter configuration c
  - set of system properties $\mathcal{P} = \{p_1, \ldots, p_n\}$
  - set of (user defined) weights $w_1, \ldots, w_n$
  - real number k
- We define:

$$SDR_{\mathcal{P};c} = \begin{cases} \sqrt[w]{\prod_{i=1}^{n} \mid \text{slack}_{p_i;c}^{w_i} \mid} & , \; k = 0 \\ \sqrt[k]{\frac{1}{w} \times \sum_{i=1}^{n} \left(w_i \times \mid \text{slack}_{p_i;c}^{k} \mid \right)} & , \; otherwise \end{cases}$$

- Where: $w = \sum w_i$

Static Design Robustness (3)

- Impact of *k* on the SDR metric value for two design properties p_1 and p_2



Dynamic Design Robustness (1)

- robustness potential of a system with respect to the variation of a specific design property
- includes potential counteractions in reaction to design property variations
  - Scheduling parameter adaptation, application remapping, etc.
- Design scenario:
  - parameters can be modified during product life time or in the field

Dynamic Design Robustness (2)

- Given:
  - constrained system S
  - design property p
  - set of potential parameter configurations $C = \{c_1, \ldots, c_n\}$
- We define:

$$DDR_{p;C} = \|\mathcal{V}\|_\infty = \max_{c \in C} |\text{slack}_{p;c}|$$

- DDR is not unique but depends on the set of available configurations ("counteractions") in C