Effects of Design Space Discretization on Constraint Based Design Space Exploration

Ludwig Karlsson
ludwigka@kth.se

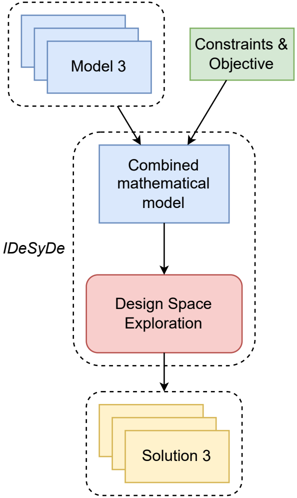
This project was done in collaboration with Saab

Outline

- **Background**: Design Space Exploration, Multiresolutional analysis
- **Method**: Models, Experiments
- **Analysis**: Convergence
- **Results**
- **Discussion**
- **Summary**

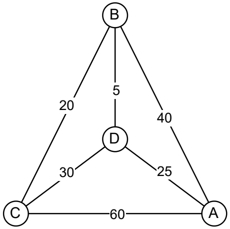
Background: Design Space Exploration (DSE)

- **Exploring** a space of designs
- Embedded systems design
- **Optimization**
- Requires **models** and an **optimization procedure**
- **Modern platforms** have made the design space **highly complex**



Background: Constraint Programming (CP)

- Programming paradigm for combinatorial problems
- A way to formulate **models**
- Set of **decision variables** with corresponding **domains**
- Set of **constraints** that variables must satisfy
- Objective function
- Efficient solvers typically require **integer domains**
  $\implies$ **discretization**



Background: Multiresolutional analysis (MRA) [1]

- A family of **nested subspaces** $V_0 \subset V_1 \subset V_2 \subset \cdots \subset L^2([0,1])$
- Orthonormal bases $V_m = \mathrm{span}\{\phi_{mk}\}_{k=0}^{2^m-1}$
- *E.g.*, spaces of **piecewice constant functions**
- Allows studying **differences** between spaces: $V_{m+1} = V_m \oplus R_m$
- Bases for R_m can be constructed from ϕ_{mn}: **multi-wavelet bases**
- **Error bound**: $\|Q_m f - f\| \leq \frac{1}{2^{m+1}} \sup_{x\in[0,1]} |f(x)|$

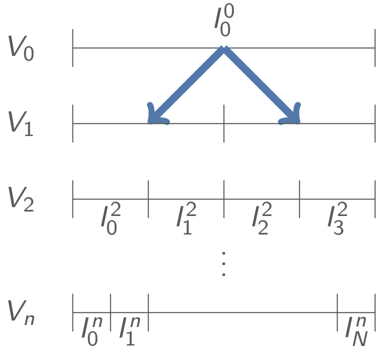


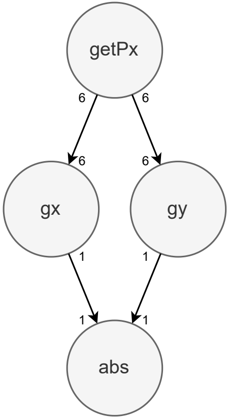
Figure used and modified with permission by Jennifer Ryan.

Method: DSE problem

- **Mapping** and **scheduling** of streaming-applications onto tile-based platforms
- Application model: **Synchronous Data Flow** (SDF)
- Platform model: **Tile-based** predictable time-division multiplexing bus-based **multiple-processor** system-on-chip architecture
- Used in previous research [2] [3]
- **IDeSyDe**: DSE tool used to combine the models (DSI) and then explore the resulting space [4]

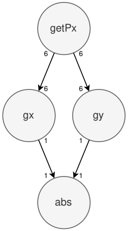
Method: Synchronous Data Flow (SDF) [5]

- Modelling framework for **streaming applications**
- Directed **graph** with loops: actors and channels
- Data are modelled as "**tokens**"
- An actor consumes and produces a certain number of tokens when it "**fires**"
- An actor may fire **immediately** when sufficient tokens are available
- Three applications: Sobel filter (**S**), RASTA-PLP (**R**), JPEG-Encoder (**J**)

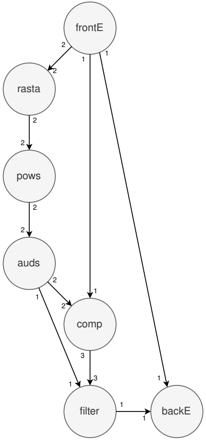


Sobel (S)

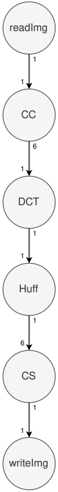
Method: SDF Applications



Sobel (S)



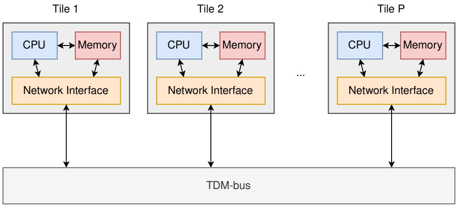
RASTA-PLP (R)



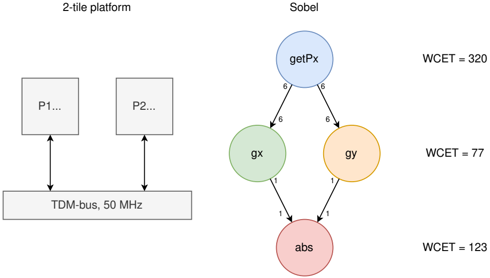
JPEG-encoder (J)

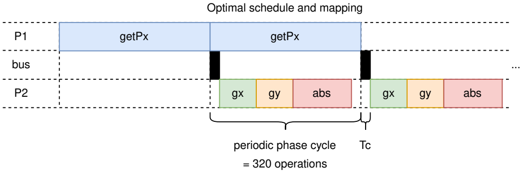
Method: Platform model [3]

- $|P|$ identical and **independent** tiles connected by a **single bus**
- Each tile contains a single-core processor, memory, and a network interface
- At any time, data may be sent between a **single** pair of tiles through the **bus**
- **Cyclic order** elements



Method: DSE Problem





Method: Mathematical formulation

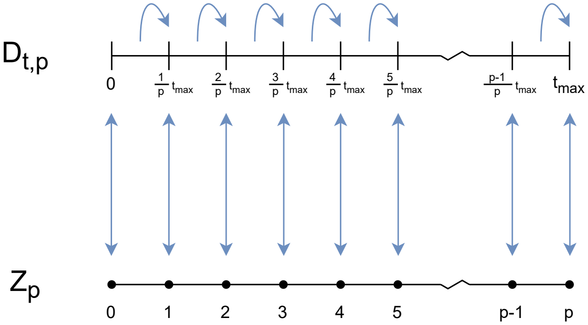
- Application- and platform models expressed with CP
- Objective: **Throughput** Th measures the number of **actor firings per time unit** in the solution
- **Discretization**: Time (and memory) is discretized according to a parameter p_t
- **Discrete throughput** $\mathrm{Th}(x; p)$ used in optimization
- Taking $p_t = 2^q, q \in \mathbb{N}$ gives $D_{t,2^{q+1}} \subset D_{t,2^q}$

Method: Mathematical formulation

- **Discretization**: Time is discretized according to a parameter p_t:

$$D_{t,p_t} = \left\{ \frac{k t_{\max}}{p_t} : k \in \mathbb{N}_{0,p_t} \right\}$$

- Values between the discrete points are **rounded upwards**
- $t_{\max}$ is computed as a worst-case total time

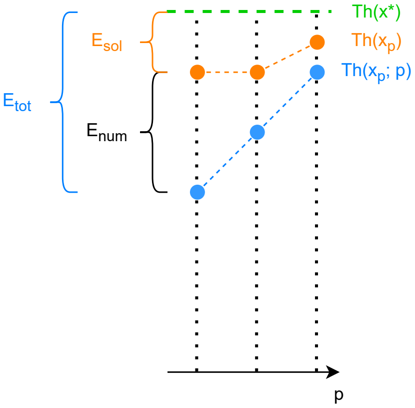


Method: Experiments

- Models: S2, S4, R4, RJ3, SRJ2
- A particular set of time discretizations P_t
- Each point evaluated by **total error** E_{tot} and **solution error** E_{sol}:

$$E_{\mathrm{tot}} = \mathrm{Th}(x^*) - \mathrm{Th}(x_p^*; p)$$

$$E_{\mathrm{sol}} = \mathrm{Th}(x^*) - \mathrm{Th}(x_p^*)$$



Analysis: Convergence

Theorem: *The numerical throughput of the optimal equivalence class* $[x^*_{2^q}]$ *in the approximation spaces converge to the exact throughput of the optimal equivalence class* $[x^*]$ *in the mixed-integer space:*

$$\lim_{q\to\infty} \max_{x_{2^q}\in D_{2^q}:C} \mathrm{Th}(x_{2^q}; 2^q) = \max_{x\in D:C} \mathrm{Th}(x)$$

strictly and with a convergence speed of order 1. Specifically, there exists a problem-specific constant $C \in \mathbb{R}^+$ *such that the total error* $E_{\mathrm{tot},2^q}$ *at resolution* 2^q *is bounded by*

$$E_{\mathrm{tot},2^q} \leq C \cdot 2^{-q+1},$$

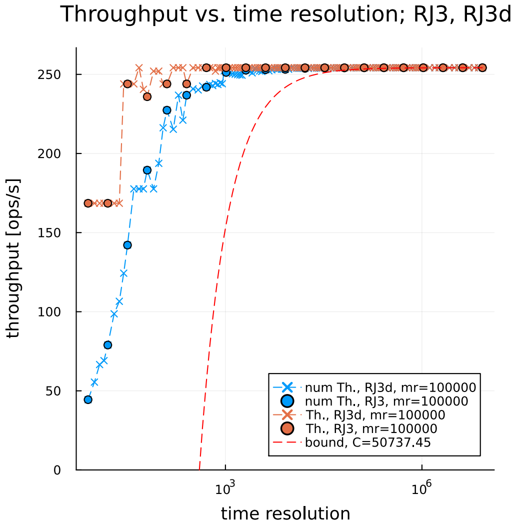
where $C \leq \mathrm{Th}(x^*)^2|\mathcal{J}|t_{\mathrm{max}}(\Gamma + 1)$.

Analysis: Convergence

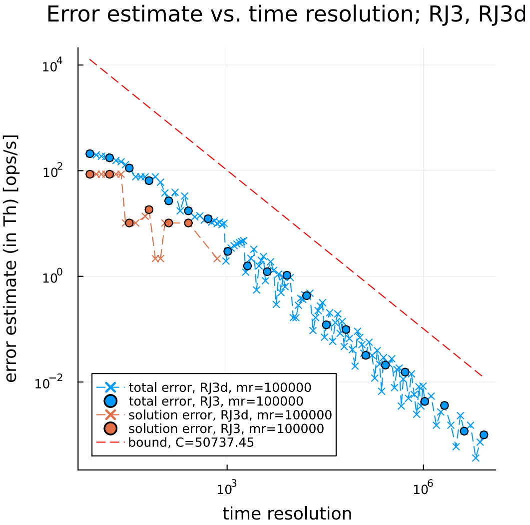
Theorem:

- The method **converges** to the exact solution
- The **total error is bounded** by $E_{\mathrm{tot},2^q} \leq C \cdot 2^{-q+1}$
- The convergence is of **order 1**
- The parameter C can be bounded from **properties of the problem instance**

Results: RJ3

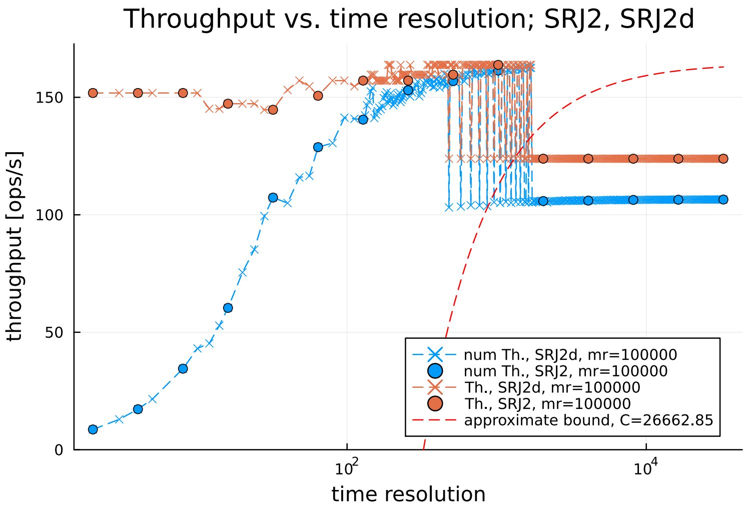


Throughput vs. time resolution; RJ3, RJ3d
throughput [ops/s]
time resolution
250
200
150
100
50
0
10^3
10^6
num Th., RJ3d, mr=100000
num Th., RJ3, mr=100000
Th., RJ3d, mr=100000
Th., RJ3, mr=100000
bound, C=50737.45



Error estimate vs. time resolution; RJ3, RJ3d
error estimate (in Th) [ops/s]
time resolution
10^4
10^2
10^0
10^{-2}
10^3
10^6
total error, RJ3d, mr=100000
total error, RJ3, mr=100000
solution error, RJ3d, mr=100000
solution error, RJ3, mr=100000
bound, C=50737.45

Results: SRJ2



Discussion

- All experiments **converge** with a speed of order 1
- All experiments satisfy the **error bound**
- The theorem **proves convergence** of the scheme
- The error bound is also **practically useful**
- The error bound could maybe be made stricter
- The analytical treatment should be applicable to **other application- and platform models**

Summary

- A discretization scheme for DSE with CP was **investigated**
- The discretization scheme used by, *e.g. IDeSyDe*, **converges** to the correct solution
- Convergence is of **order 1**
- This was proven **analytically** and verified **experimentally** for a particular choice of models

Sources

[1] B. K. Alpert, "A class of bases in l2 for the sparse representation of integral operators," *SIAM journal on mathematical analysis*, vol. 24, no. 1, pp. 246–262, 1993.

[2] K. Rosvall and I. Sander, "A constraint-based design space exploration framework for real-time applications on mpsocs," in 2014 *Design, Automation & Test in Europe Conference & Exhibition (DATE)*. EDAA, 2014. ISBN 9783981537024. ISSN 1530-1591 pp. 1–6.

[3] K. Rosvall and I. Sander, "Flexible and tradeoff-aware constraint-based design space exploration for streaming applications on heterogeneous platforms," *ACM transactions on design automation of electronic systems*, vol. 23, no. 2, pp. 1–26, 2018.

[4] R. Jordão, "IDeSyDe: Design space identification and exploration," Accessed: 2023-05-31. [Online]. Available: `https://forsyde.github.io/IDeSyDe/`

[5] E. Lee and D. Messerschmitt, "Synchronous data flow," *Proceedings of the IEEE*, vol. 75, no. 9, pp. 1235–1245, 1987.

Thanks!