Combined Data Transfer Response Time and Mapping Exploration in MPSoCs

WATERS 2019 Industrial Challenge

Co-located with 31st Euromicro Conference on Real-Time Systems (ECRTS)

July 9, 2019

Stuttgart, Germany

SPONSORED BY THE

Federal Ministry of Education and Research

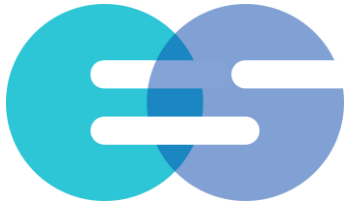
Alexander Diewald, Simon Barner

fortiss GmbH, Research Institute of the Free State of Bavaria, Munich, Germany

fortiss

Selma Saidi

Institute of Embedded Systems, TU Hamburg, Germany



Selected Focus on WATERS 2019 Challenge

Contributions

- Combined Data Transfer Response Time and Mapping Exploration in MPSoCs
  - Explore task mappings to an heterogeneous platform
  - Task response time computation considers DMA transfers and data prefetching
  - Minimize completion time

- Goals
  - Fast exploration for early design-phases considering MPSoC characteristics
  - Accurate results

Selected Focus on WATERS 2019 Challenge

Additional Assumptions (on Top of Challenge Specification)

- **Assumptions**
  - All tasks execute according to AER model: Acquisition (read), Execution, Restitution (write) phase
  - Suppose that all memory accesses are performed by DMA requests
  - DMA data prefetching serializes requests: No interference effects on latency
  - Assume that the bandwidth of the DMA accesses equals those of the GPU
  - Do not consider task periods nor preemption

- **Rationale**: Integration of Model-Driven Engineering and Design-Space Exploration (DSE)
  - Optimize task mapping and scheduling
  - Consider data transfers (data fetching + prefetching)

- **Approach**: Integration of approach into AutoFOCUS3 MDE tool
  - Complete flow from the provided AMALTHEA model to optimized schedules
  - Basis for experimental validation

Optimization Problem

- Problem Size
  - # Tasks: 39 (14 actual tasks + 25 R/W tasks)
  - # Cores: 9 (incl. 2x DMA)
  - Allocation Choices: ~1.64 * 10e37;
    Considering alloc. constraints: ~6.05 * 10e7;

	Denver	A57	GP10b	DMA
Lidar_Grabber	10868	13660		
READ_Lidar_Grabber				35
WRITE_Lidar_Grabber				94
Localization	294808	387420	124000	
READ_Localization				70
WRITE_Localization				0.33
Detection			116000	
READ_Detection				23
WRITE_Detection				65

WCET & R/W Latencies in µs (excerpt)

- Problem Formulation

$$\min_{\forall e_j \in \mathcal{E}} \max_{\forall \tau_i \in \mathcal{T}^{e_j}} (t^{e_j}_{end,i})$$

$$s.t. \quad \boldsymbol{u}_i^T \boldsymbol{A} \boldsymbol{a}_i = 1, \qquad \forall \tau_i \in \mathcal{T},$$

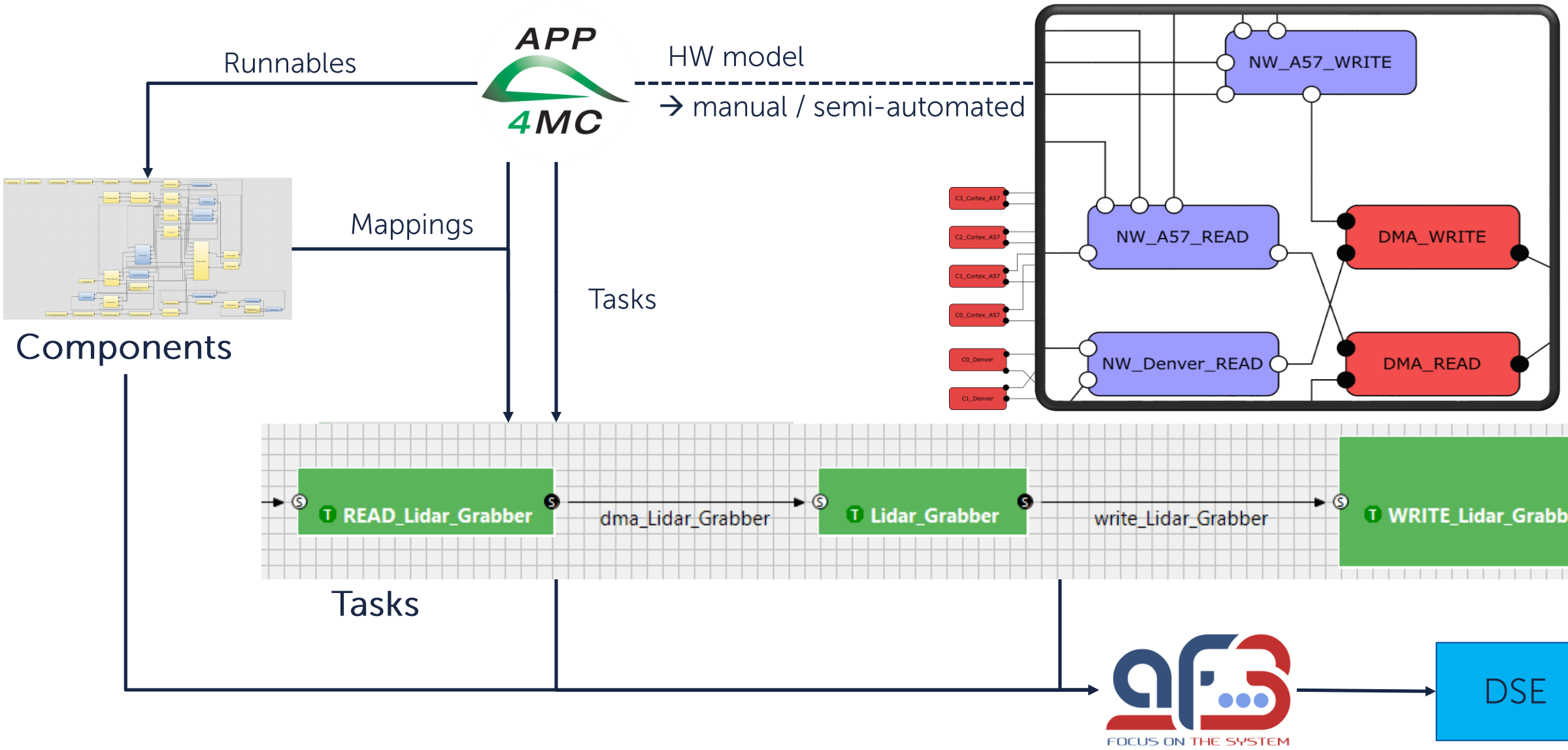
$$\exists_{r \in \mathcal{R}} r\left(f_{send,a}(m_i)^T \boldsymbol{e}, u_j^T f_{recv,a}(m_i) \boldsymbol{e}\right), \qquad \forall m_i \in \mathcal{M}, \forall j \in \{1, \dots, R\}$$

$$t_{end,j} \leq t_{start,i}, \qquad \forall \tau_j \in \mathcal{T}_{pred,i}, \forall \tau_i \in \mathcal{T}$$

$$\dots$$

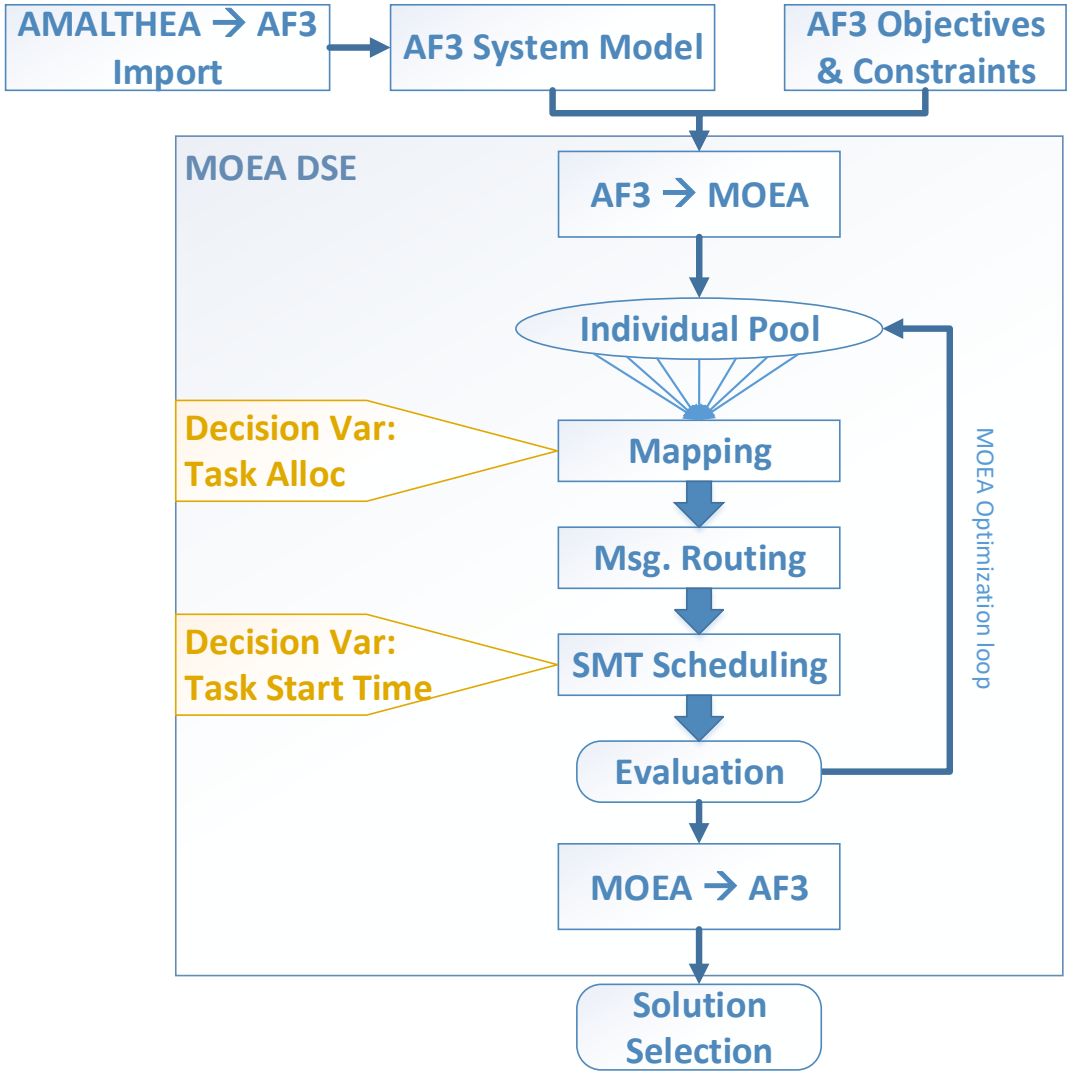
➔ Prefetching is achieved by implicit parallelism of non-dependent tasks and the AER model

Overview: Model-Driven Engineering



Overview: Design Space Exploration

- MOEA DSE integrates in AF3
- Main loop: Use MOEA to explore task-to-core mappings
  - Benefit from flexibility and performance of MOEA generic optimization problems
  - Here: Optimize mappings w.r.t. resulting schedule latencies
- SMT-based schedule synthesis (Z3)
  - WCETs depending on allocation to heterogeneous architecture
  - Memory transfers: Estimate latencies using R/W tasks

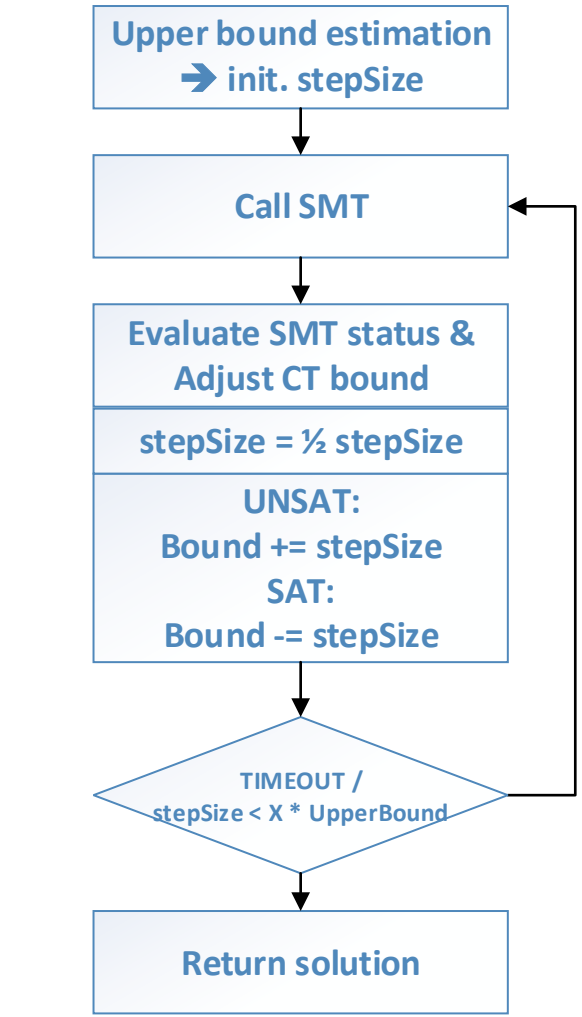


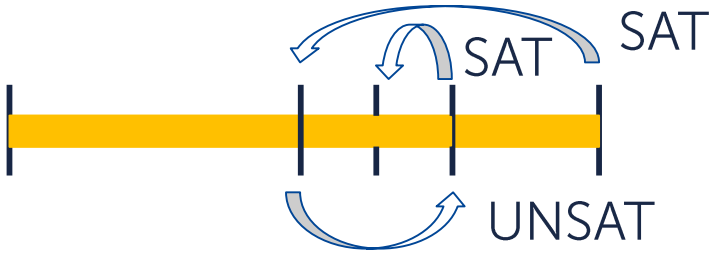
SMT Scheduler Algorithm

- Bisection algorithm
  - Unsuitable performance of the automatically activated Z3 optimizer
    Disclaimer: The large set of Z3 strategies was not intensively examined
  - Minimize upper bound for schedule completion time
  - Initial upper bound: all tasks are allocated to a single core (worst case)

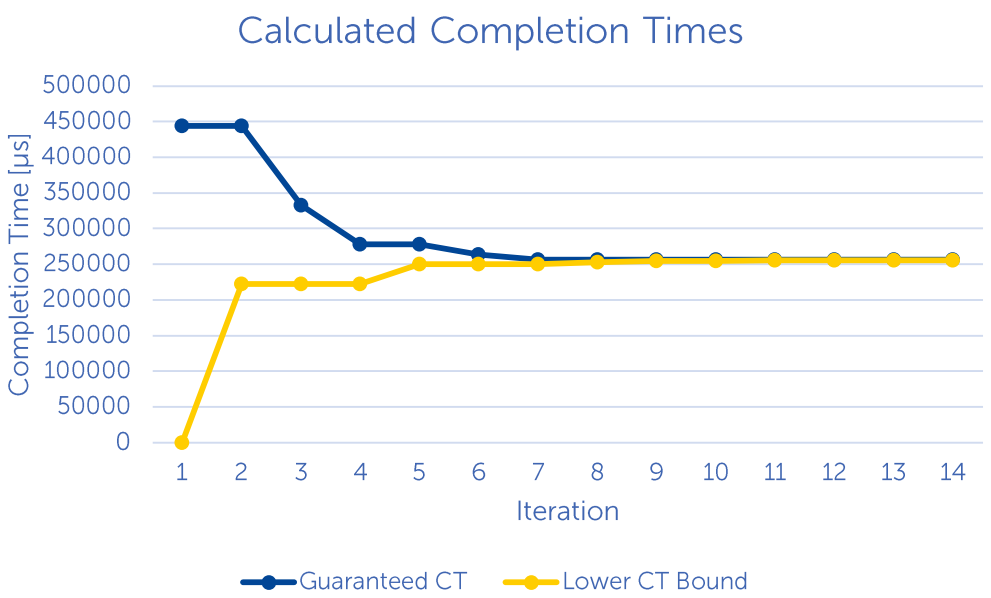
- Use SMT status information (SAT, UNSAT) to determine
  - Next step's direction
  - Solution's schedule completion time accuracy based on UNSAT results that provide a lower bound

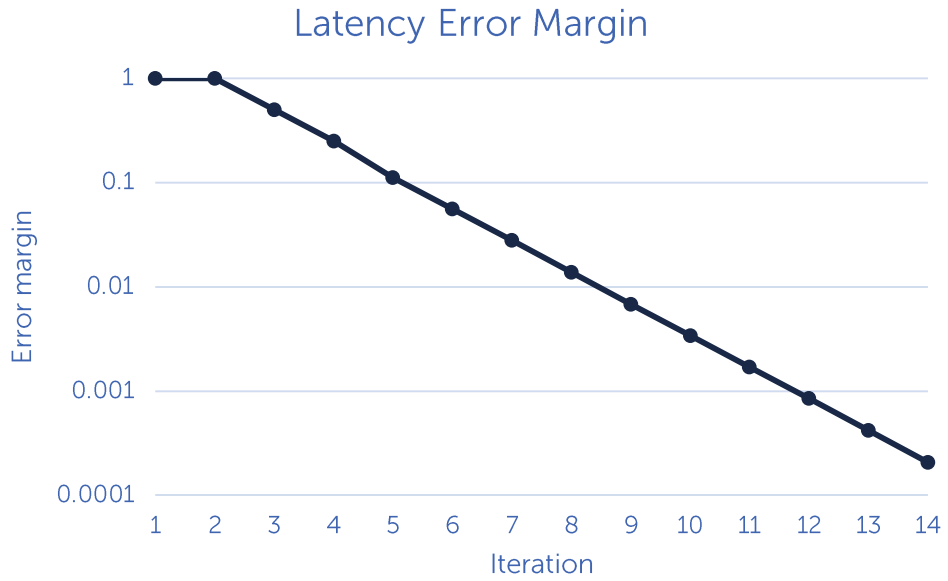
- Termination criteria
  - stepSize < X * *UpperBound, or*
  - SMT timeout to bound exploration runtime: Running time increases exponentially if bound is close to the optimum





Approaching the Optimal Schedule Completion Time

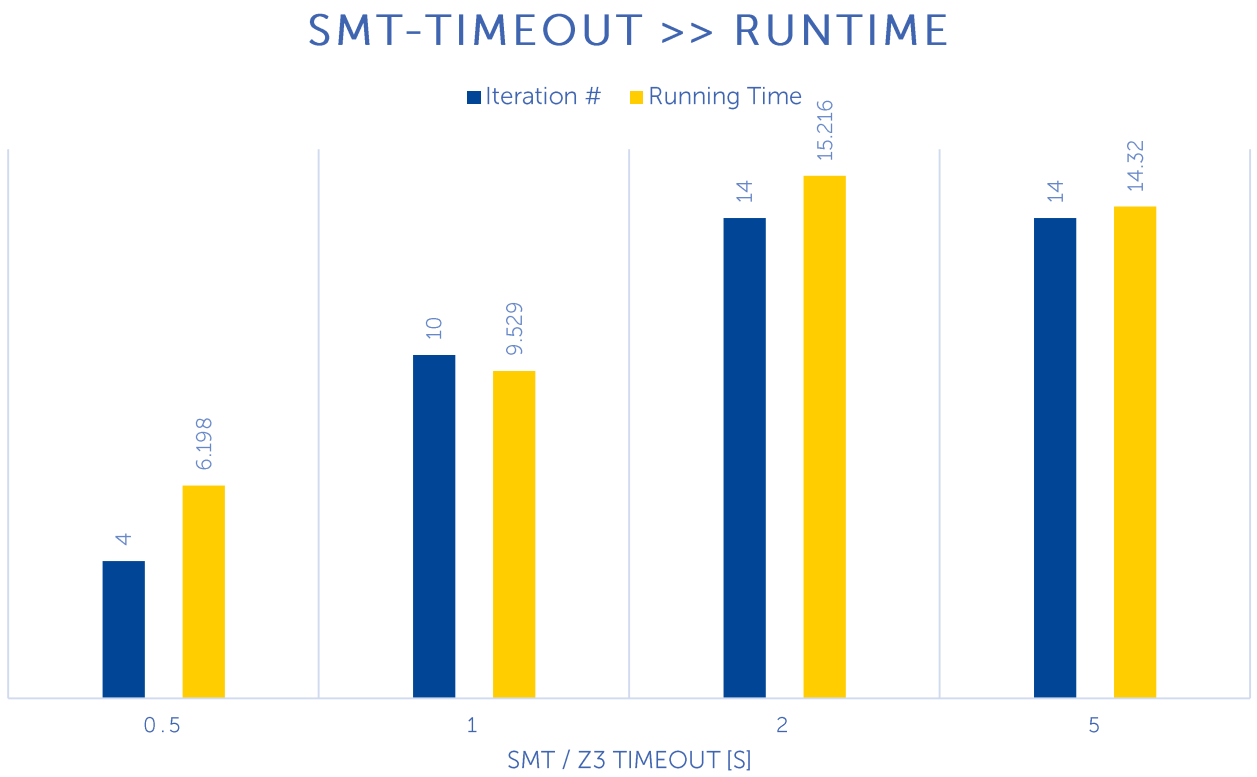


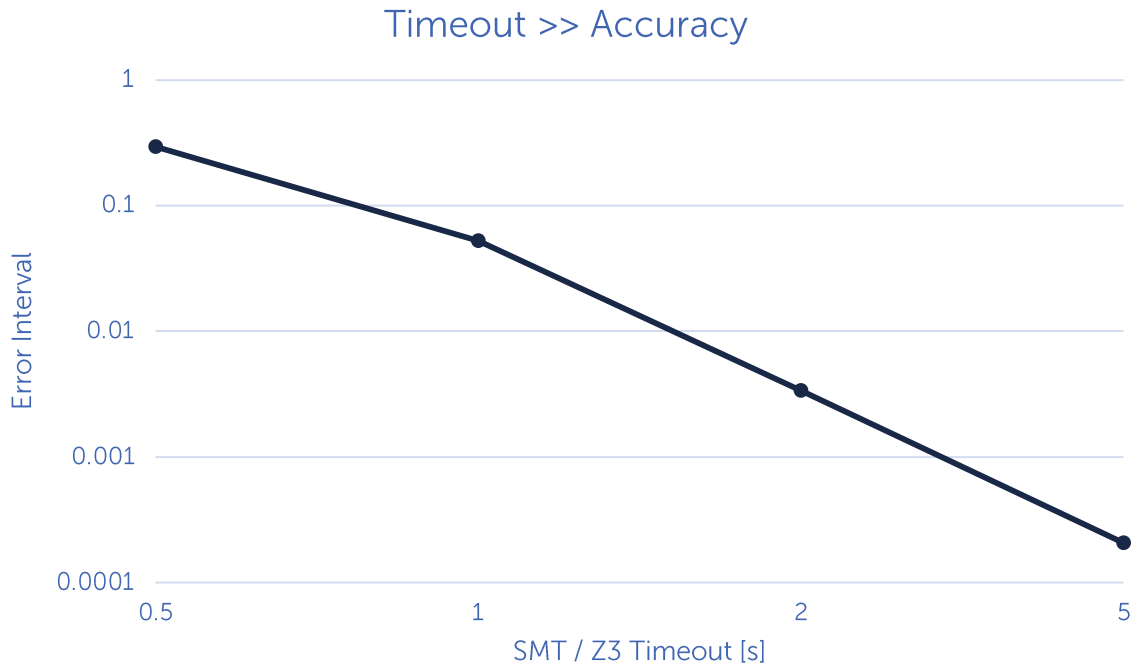


- Roughly 4-5 iterations are needed for reasonable latency estimates
- In iteration 14, the close-to-optimal completion time is found (minimum step size reached)

- Error margin: difference between guaranteed completion time (SAT) & lower bound (UNSAT)
  - Logarithmic dependency (bisection)
  - Fast accuracy increase per iteration

SMT Timeout vs. Accuracy





- Termination criterion: stepSize < 10e-4 * *UpperBound*
- Number of iterations ~ runtime
- Accuracy increases rapidly

Decreasing the SMT / Z3 timeout

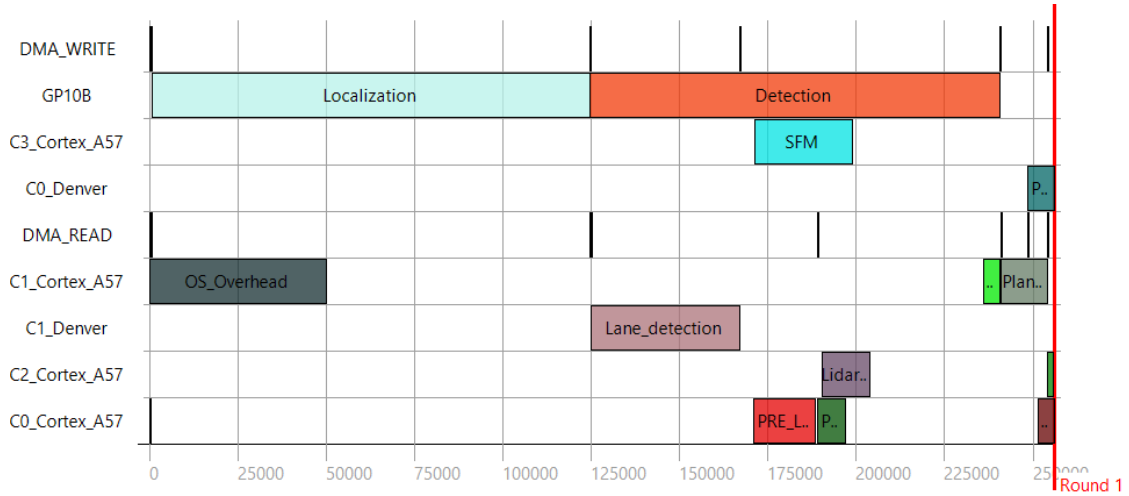
	DSE Running Time			Maximum Solution Error		
Timeout	Min	Avg	Max	Min	Avg	Max
5s	12.371	14.2238	16.767	2.07E-04	2.07E-04	2.07E-04
2s	12.614	14.4496	17.862	2.07E-04	2.07E-04	2.07E-04
1s	6.713	9.0222	11.069	1.69E-03	9.56E-02	3.22E-1

- Experiment: 5 executions of bisection algorithm per SMT timeout (1s, 2s, 5s)
  - Reproducible running time of SMT scheduler for the same problem instances
  - Large variance of accuracy for a 1s timeout
  - Worst case running time for 1s and 2s experiments are similar

- Larger timeouts provide better accuracy
  - This holds until an accuracy limit is reached from which the solution will not improve
  - Goal: Find sweet spot between DSE running time and accuracy (here: 2s)

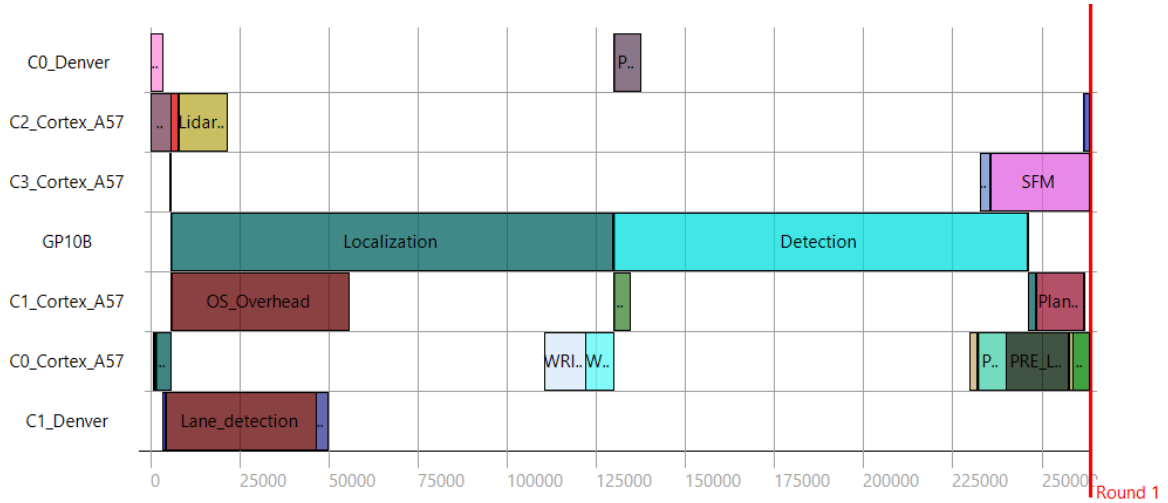
Effects of DMA Prefetching: Task Allocation 1

- Parameters
  - SMT Timeout: 5s
  - Termination Criterion:
    StepSize < 10e-4 * *UpperBound*



DMA & Prefetching

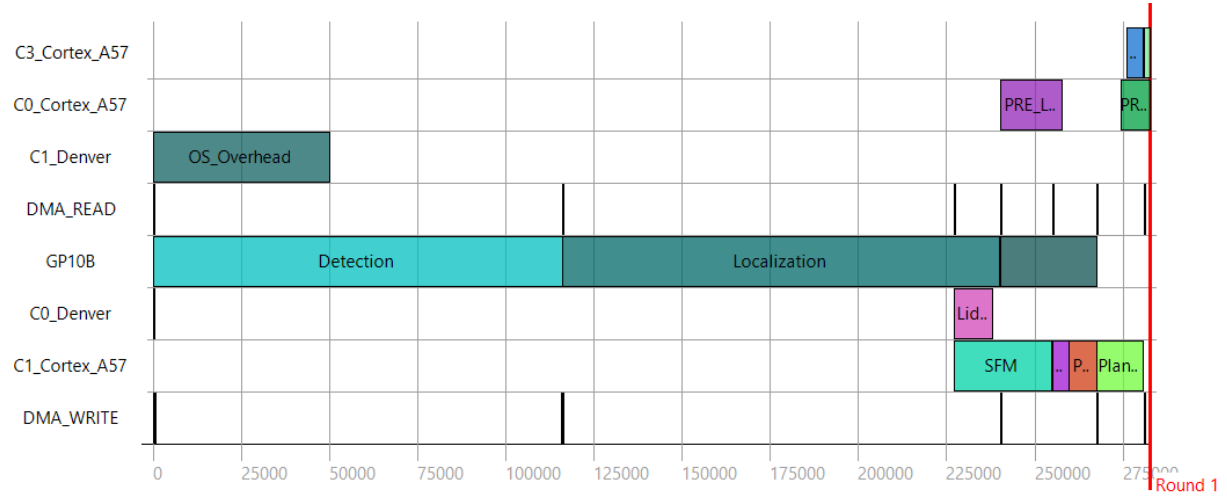
- Completion time (Upper Bound)
  - DMA & prefetching
    256071 µs
  - No DMA & prefetching
    263498 µs
  - Relative improvement
    2.82 * 10e-2



No DMA & Prefetching

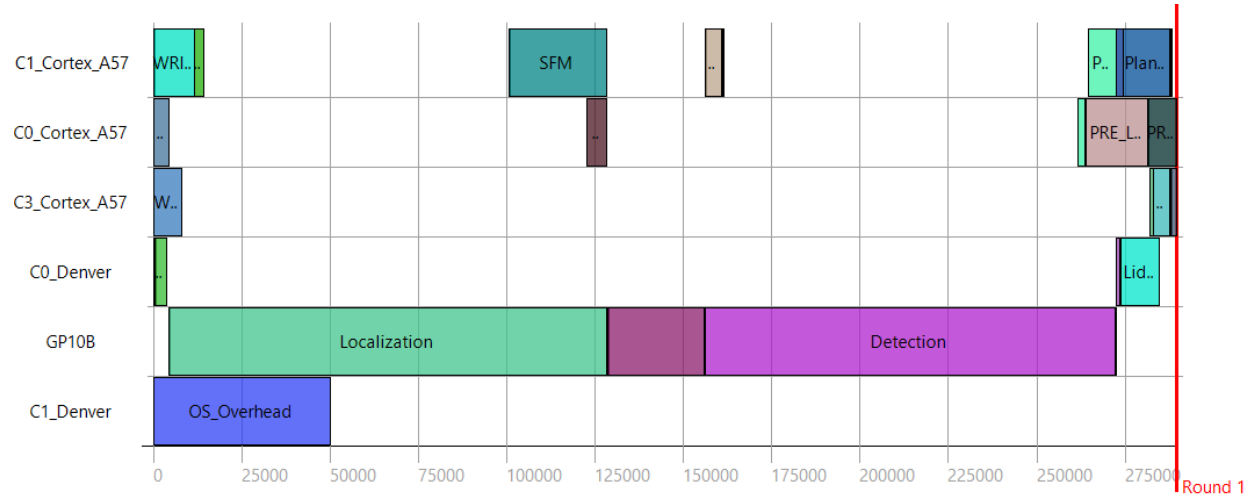
Effects of DMA Prefetching: Task Allocation 2

- Parameters
  - SMT Timeout: 5s
  - Termination Criterion:
    StepSize < 10e-4 * *UpperBound*



DMA & Prefetching

- Completion time (Upper Bound)
  - DMA & prefetching
    282805 µs
  - No DMA & prefetching
    289685 µs
  - Relative improvement
    2.37 * 10e-2



No DMA & Prefetching

Conclusions

- Summary of Approach: Combined Data Transfer Response Time and Mapping Exploration in MPSoCs
  - Integrated into open source AutoFOCUS3 MDE tool
  - MOEA + SMT-based exploration algorithm

- Summary of Experimental Results
  - Prefer **accuracy** metrics as **termination criterion** over SMT timeouts
  - Small benefits of **DMA-based prefetching**
  - Schedule completion time is dominated by computation tasks

- Future work
  - Validate results (different task mappings and memory access patterns) on a real platform
  - DMA: optimize use and use more fine-grained model (e.g., bandwidth, interferences)
  - Task model: consider periods and preemption
  - Integrate more sophisticated timing analysis tool (task mapping and dependencies, and data prefetching)
  - Open question: Tune SMT timeout automatically (derive from problem size?)

Acknowledgements

This research has received funding from the German Federal Ministry of Education and Research (BMBF; grant agreement No. 01IS16025F) in the ARAMiS II project.

www.aramis2.org

Contact

Alexander Diewald, Simon Barner

fortiss GmbH
Research Institute of the Free State of Bavaria

Guerickestraße 25
80805 Munich
Germany

{diewald, barner}@fortiss.org

Selma Saidi, PhD

Institute of Embedded Systems,
Hamburg University of Technology (TUHH)

Am Schwarzenberg-Campus 3 (E)
21073 Hamburg
Germany

selma.saidi@tuhh.de