DFG

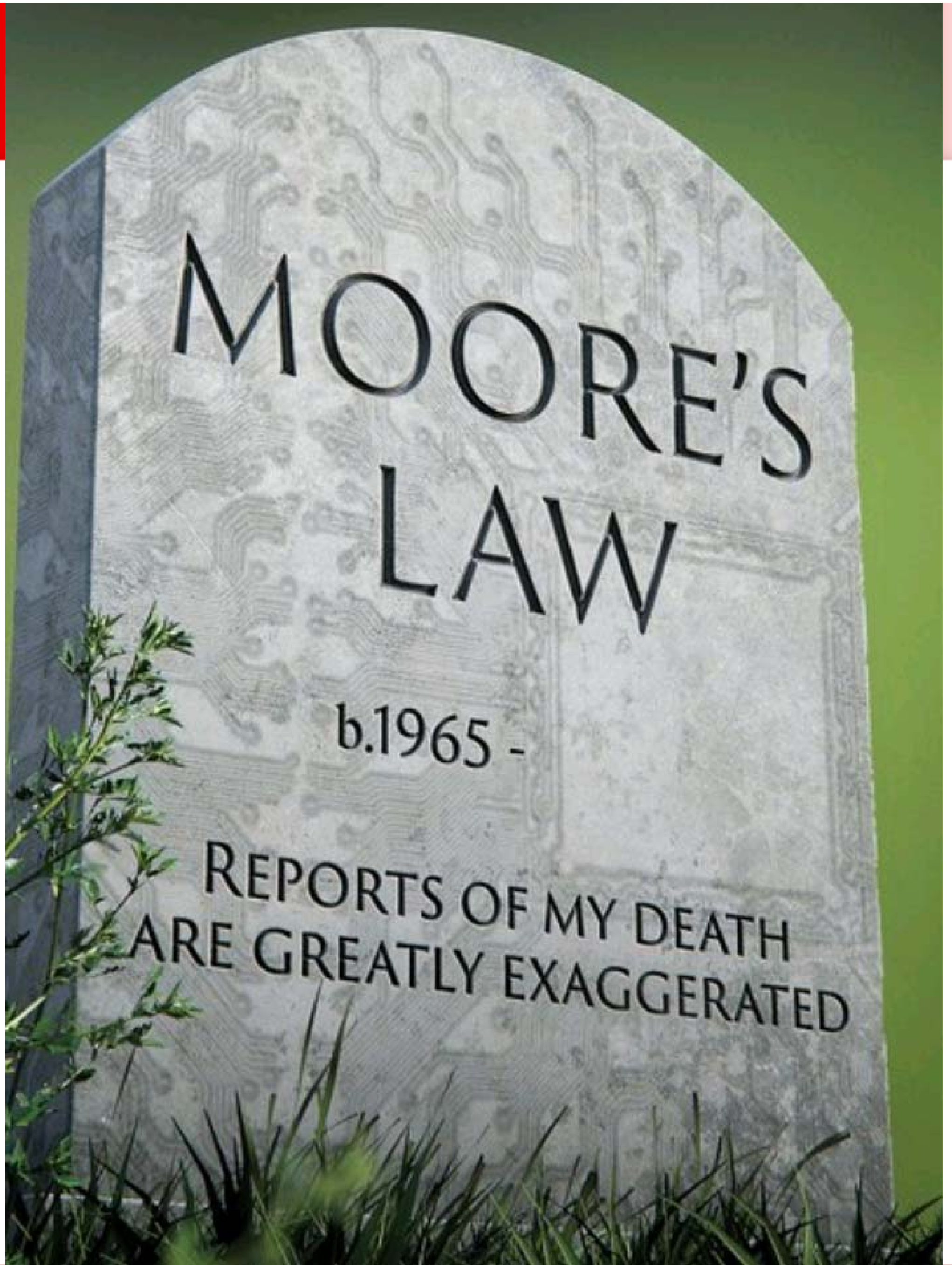
Methodologies for Application Mapping for NoC-Based MPSoCs

FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG

Karlsruhe Institute of Technology

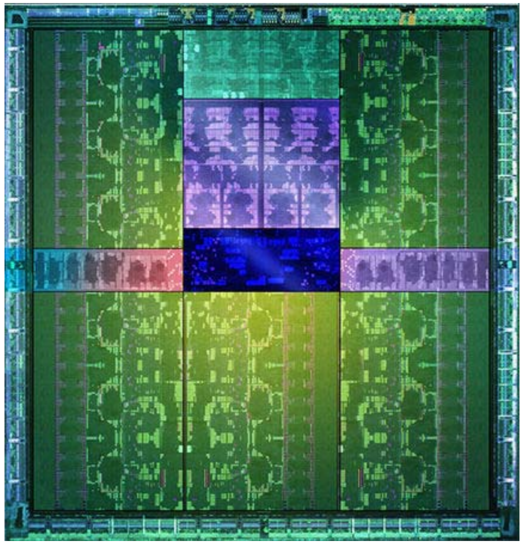
TRR 89

J. Teich, Adaptive Many-Core Architectures and Systems Workshop, York,
14 June 2018



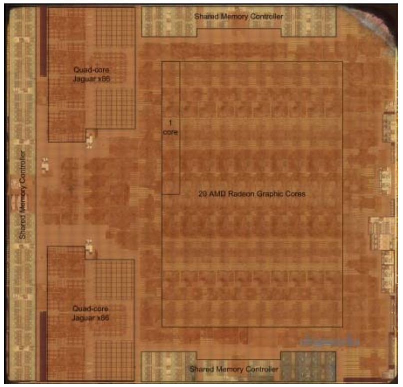
[1] ACM Communications 1/2017

Evolution of MPSoC Technology



[NVIDIA Corporation]

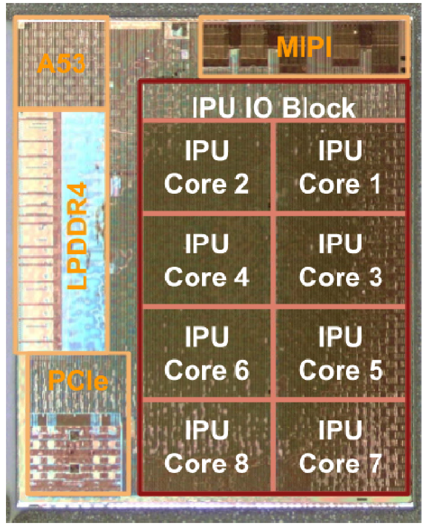
NVIDIA Kepler: 2880 ALUs



[Chipworks]

[Sony Corporation]

Sony Playstation 4
CPU: 8 cores (2x AMD Jaguar Quad-core)
GPU: 1280 ALUs (AMD Radeon HD 7870)

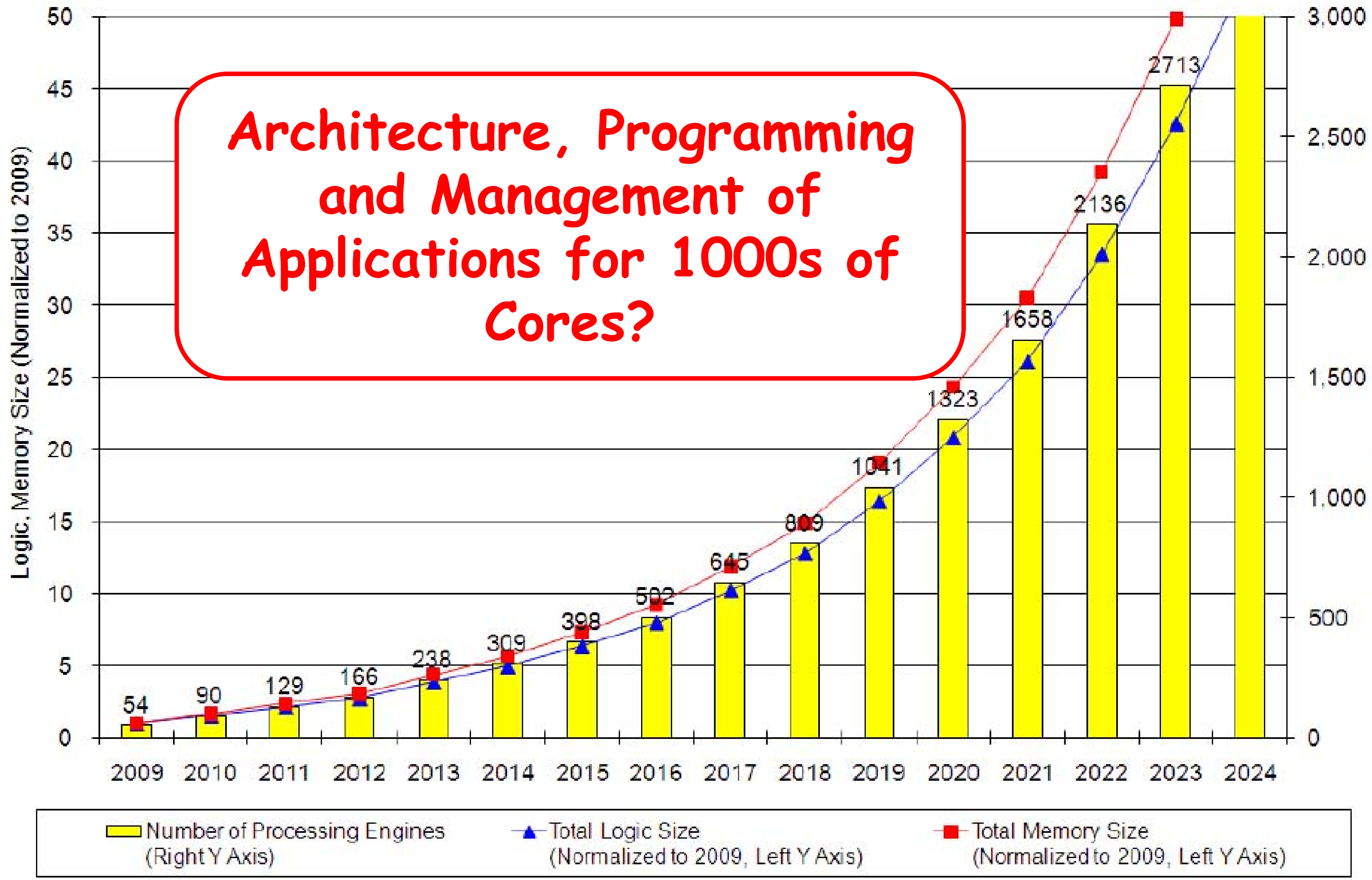


[Google Corporation]

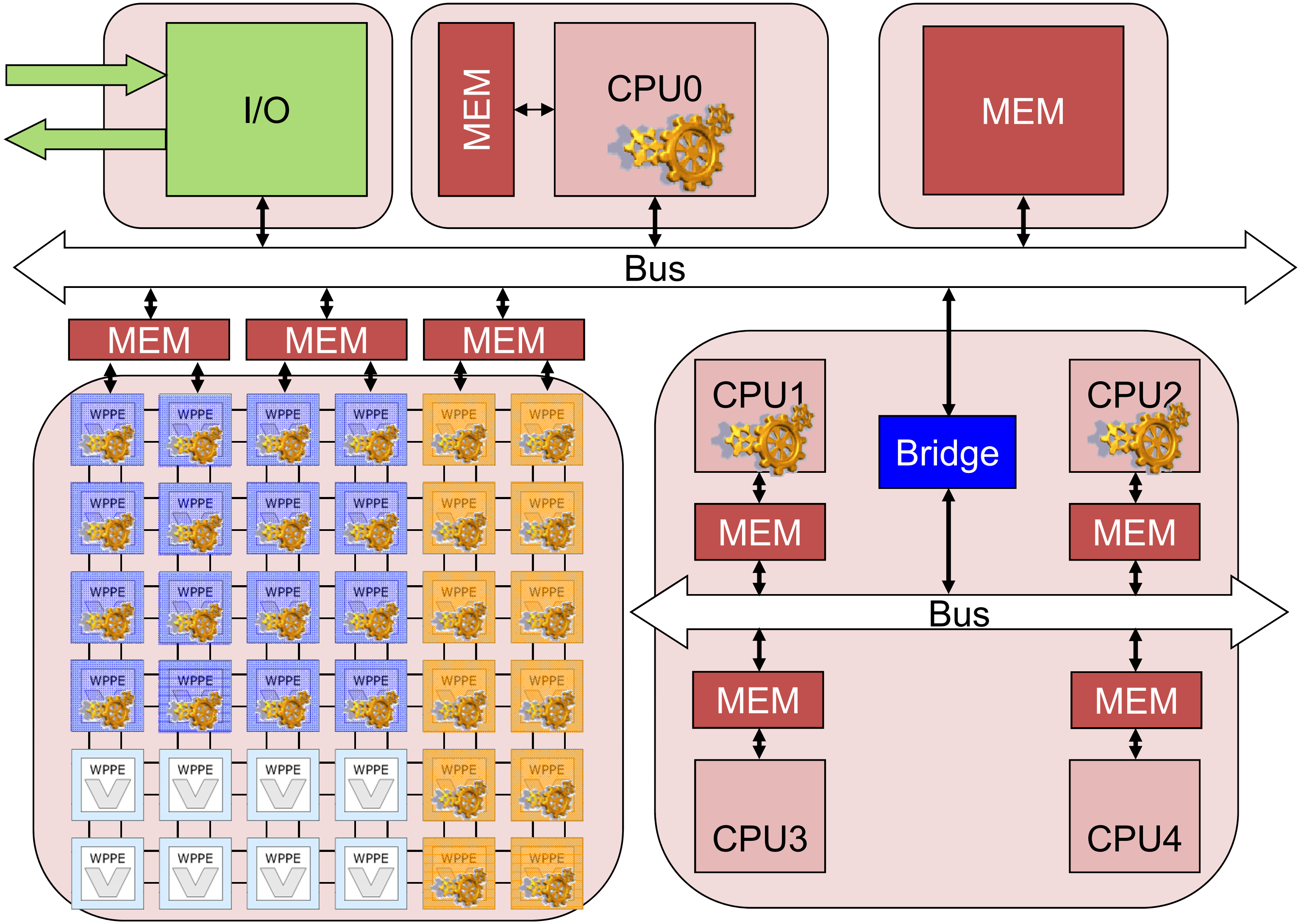
[Google Corporation]

Google Pixel 2
Visual Core
Image Processing & Machine Learning
8 x 512 ALUs; 3 TOps/s
Akku: 2700 mAh

Challenges

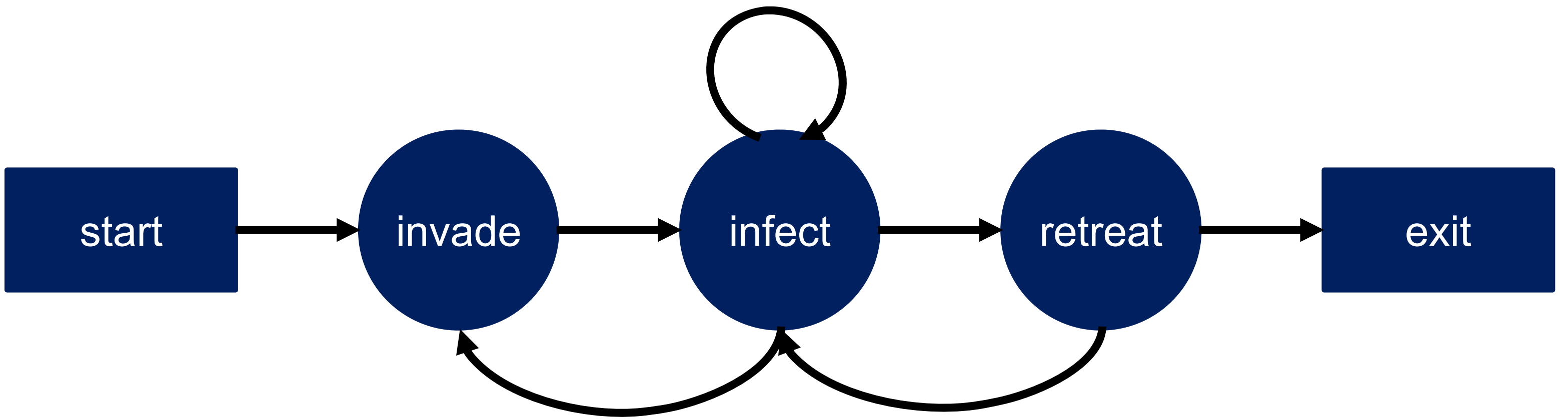


[The International Technology Roadmap for Semiconductors. International Technology Roadmap for Semiconductors 2010 Update Overview. http://www.itrs.net.]


I/O
MEM
CPU0
MEM
Bus
MEM
MEM
MEM
WPPE
CPU1
Bridge
CPU2
MEM
MEM
Bus
MEM
MEM
CPU3
CPU4

Invasive Computing

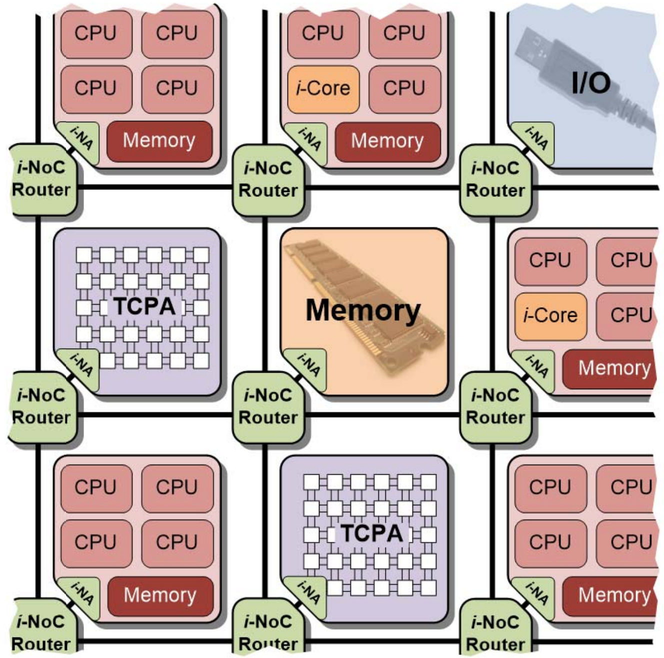
- Novel paradigm of resource-aware computing for the design and programming of future parallel computing systems
- Involves: architecture, operating systems, compiler, and algorithms research
- Three basic invasive primitives:



Application Mapping

```
// Performance Requirements
@REQUIRE("ag", new Latency(0, 110, "ms", "hard"))
val ag = new ActorGraph("ag");
/* Declare actors */
val t0 = new ImgSource("t0");
val t1 = new HSActor("t1");
val t2 = new VSActor("t2");
val t3 = new ImgSink("t3");
// Add actors and connect them
ag.addActor(t0);
ag.addActor(t1);
// ...
ag.connectPorts(t0.outPort, t1.input);
ag.connectPorts(t0.outPort, t2.input);
// ...
invade(ag);
```

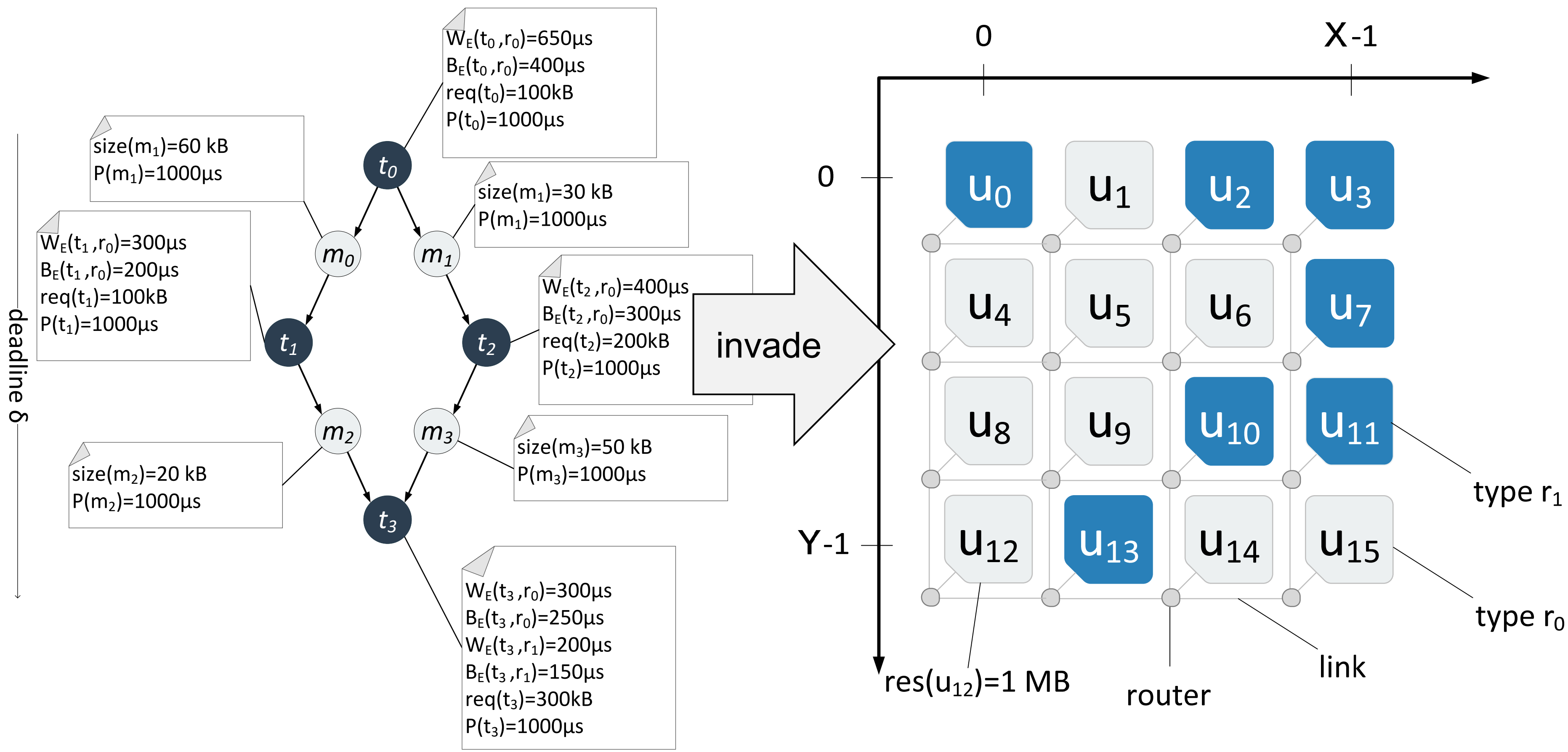
invade



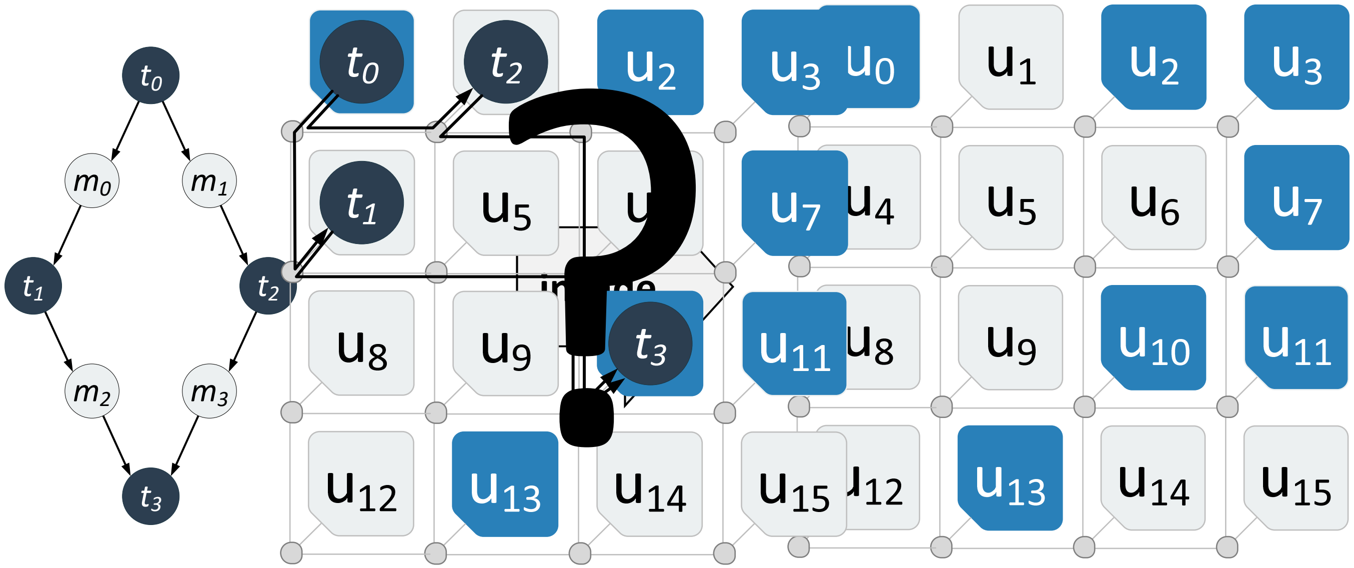
Invasive actor program [IT'16b]

Invasive NoC architecture [MOMAC'15b]

Application Mapping



Application Mapping



Application Mapping

- Optimal task mapping on heterogeneous architectures is a complex task (generalized assignment problem) [5]

- Mapping strategy highly depends on the use case:
  - One application vs. multiple applications
  - Varying application mixes vs. fixed operating modes
  - Different requirements (e.g., energy, execution time)

- Constraints have to be fulfilled, e.g.:
  - No overutilization of computational or communication resources

[5] Philip K. F. Hölzenspies, Johann L. Hurink, Jan Kuper, and Gerard J. M. Smit. Run-time spatial mapping of streaming applications to a heterogeneous multi-processor system-on-chip (MPSoC). In Proc. of DATE '08 , pages 212-217, 2008.

Design Time vs. Run Time

compute-intensive
evolutionary algorithms
s t a t i c   a n a l y s i s
DESIGN-TIME APPLICATION ANALYSIS
slow offline
e x h a u s t i v e
optimization

no guarantees
greedy
heuristics fast best effort
RUN-TIME MANAGEMENT
first fit online
dynamic
sub optimal
flexible

Overview

Self-Embedding [NOCS’11]:
Decentralized, heuristic,
fast embedding times

Hardware acceleration [MOMAC’15]

Application Mapping

Hybrid Application Mapping (HAM) [CODES’14,SORT’15,MCSOC’16, IT’16b,SpringerBook’18]:
Design-time analysis and run-time mapping

Timing [CODES’14,SORT’15, MCSOC’16,IT’16b]

Energy[CODES’14,TECS’17]

Security [SCOPES’16,IT’16a]

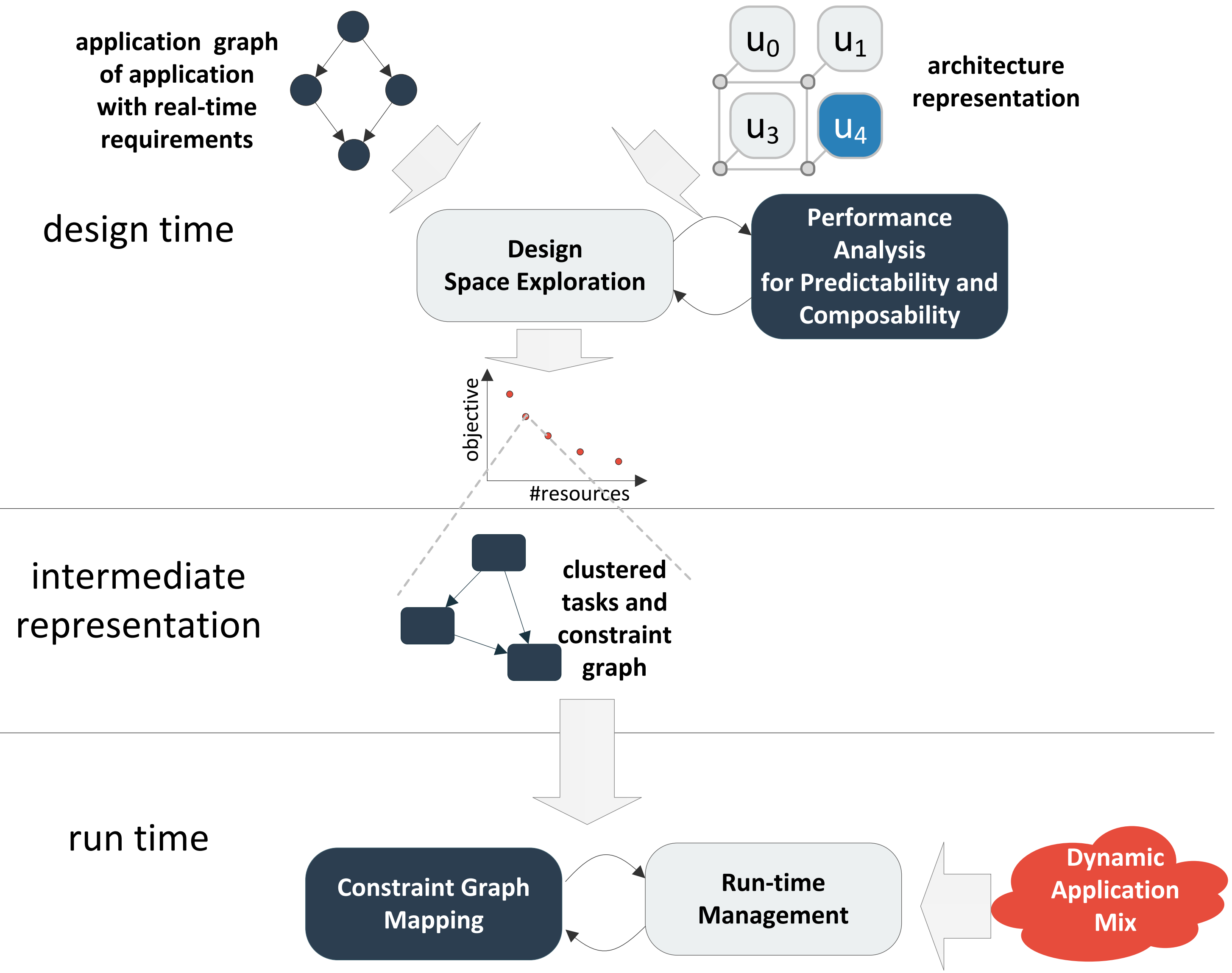
Reliability [AHS’15,MOMAC’16,IT’16c]

Hybrid Application Mapping (HAM)

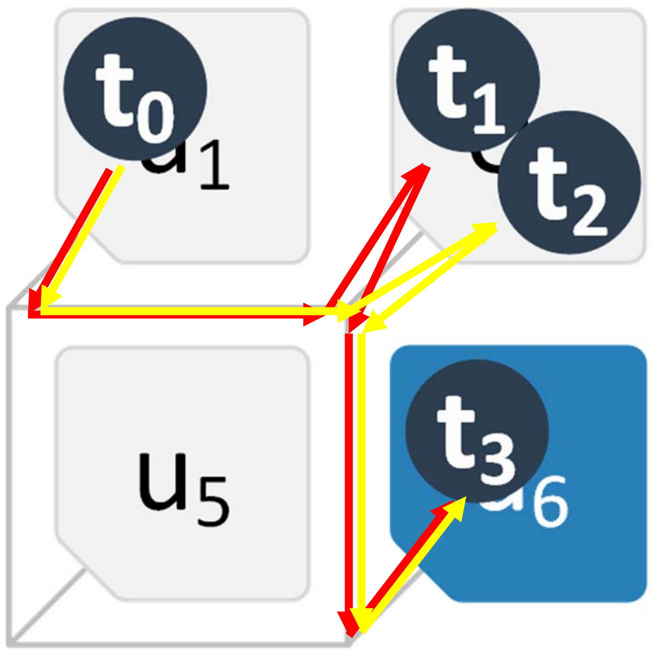
- State-of-the-art HAM approaches:
  - Scenario-based, multi-mode
  - Spatial and temporal isolation with fixed distances
- Objectives:
  - Throughput
  - Energy
- Simple communication model:
  - Dedicated point-to-point connections
  - Best effort NoCs

→ Most state-of-the-art HAM approaches are not applicable for packet-switched NoCs and are limited to certain objectives

Overview


application graph of application with real-time requirements
architecture representation
u_0
u_1
u_3
u_4
design time
Design Space Exploration
Performance Analysis for Predictability and Composability
objective
#resources
intermediate representation
clustered tasks and constraint graph
run time
Constraint Graph Mapping
Run-time Management
Dynamic Application Mix

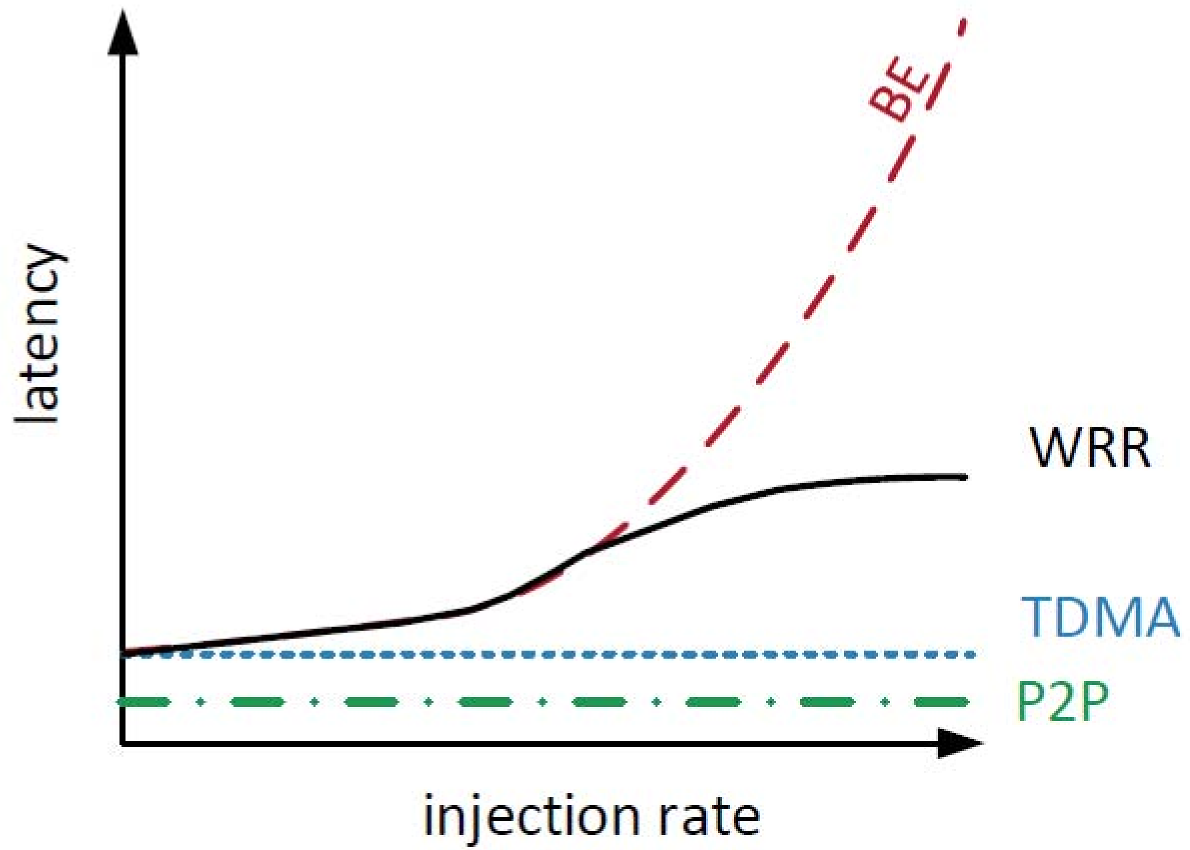
- Composability of Mapping: Each application may be analyzed independently through not sharing invaded tiles between applications and temporal isolation



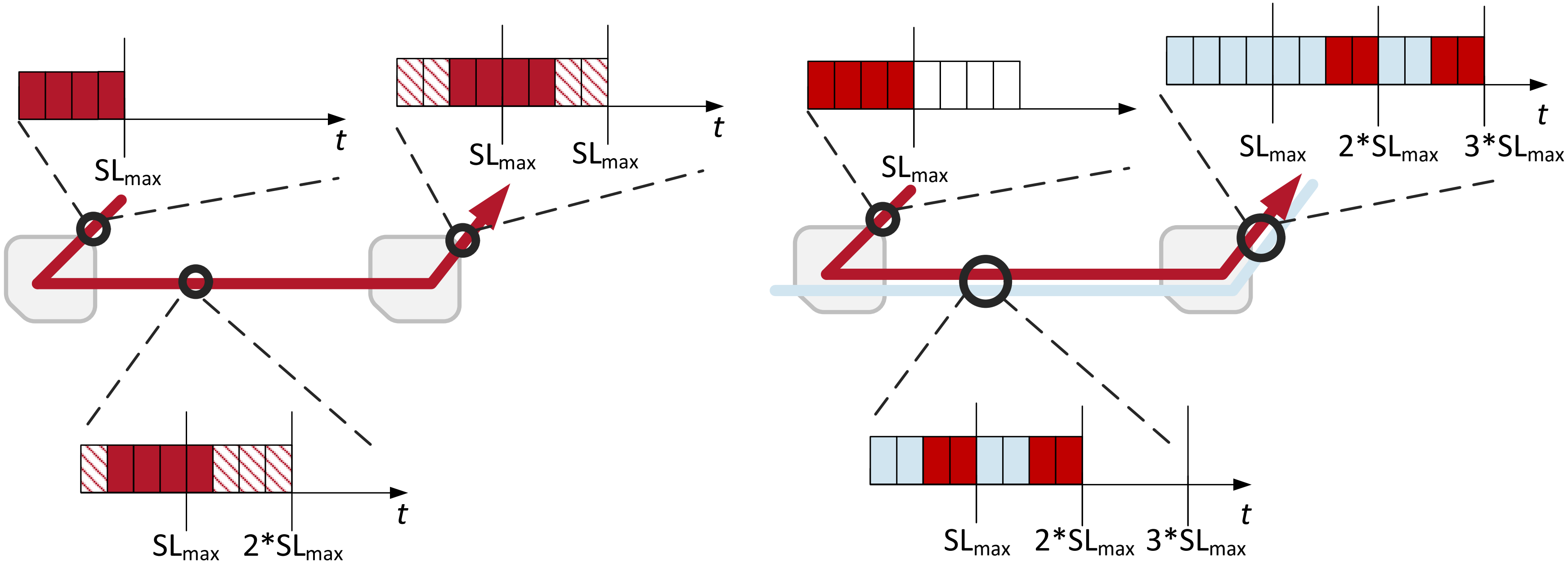
- Compositional Timing Analysis: Find the longest path and compute sum of worst-case execution latencies and worst-case communication latencies:

$$L_{path}(path, \beta, \rho) = \sum_{\forall t \in path \cap T} TL(t, \beta(t)) + \sum_{\forall m \in path \cap M} CL(m, \rho(m))$$

Why weighted round robin (WRR)?



Weighted Round Robin Arbitration of NoC Links[7]



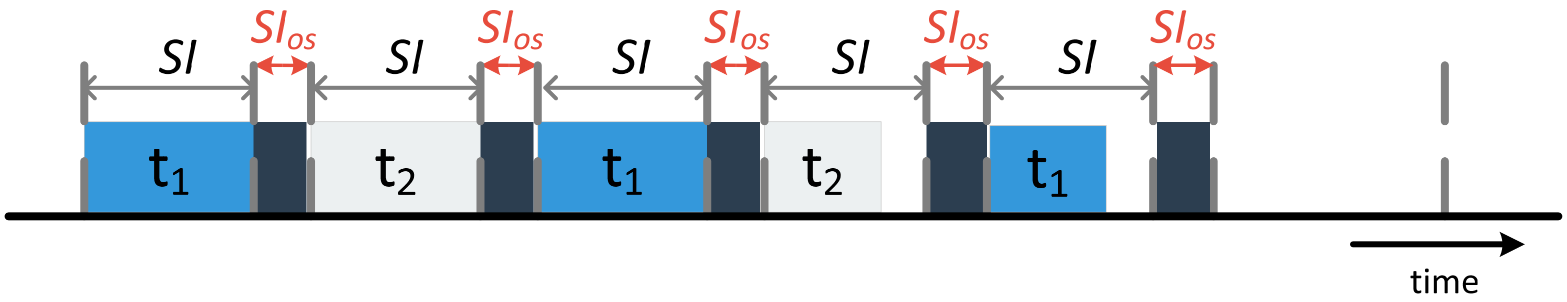
$\text{CL}^{+}(m,\rho(m)) = (n_f(m)\cdot\tau + H(u_1,u_2)\cdot\ D_R)$

$\text{CL}(m,\rho(m)) = \text{CL}^{+}(m,\rho(m)) + \left(\left\lceil\frac{n_f(m)}{SL(m)}\right\rceil - 1 + H(u_1,u_2)\right)\cdot(\text{SL}_{max}\text{-SL}(m))$

$n_f(m)$: number of flits of message m
$H(u_1,u_2)$: hop count between u_1 and u_2
D_R: Router delay
SL(m): Service Level of message m
SL_{max}: Maximum Service Level
τ: cycle length

[7] J. Heisswolf, R. König, et al. Providing multiple hard latency and throughput guarantees for packet switching networks on chip. Computers & Electrical Engineering, 39(8):2603-2622, 2013.

- The time space for scheduling tasks of the same application is broken into multiple fixed time slots, so called service intervals of length SI



- The worst-case execution latency can be divided in two parts:
  - worst-case execution time of the task without interference
  - worst-case interference from other tasks on the same CPU

$$TL\,(t,\beta(t))=TL_{exec}(t,\beta(t))+TL_{inter}(t,\beta(t))$$

CPU Execution Time Analysis

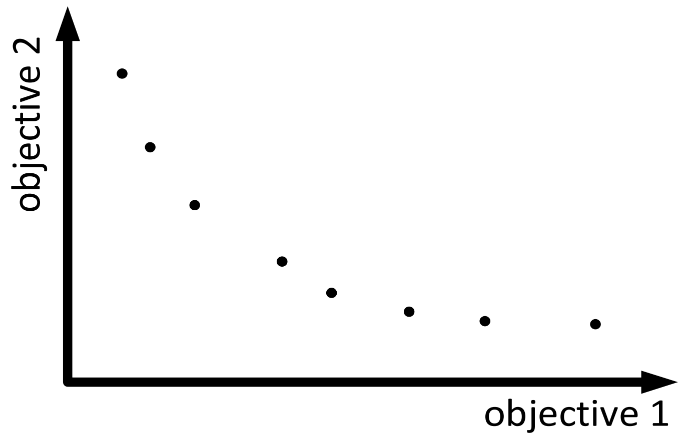
- Worst-case execution latency of task *t*:
  - $TL_t = TL_{wcet,t} + TL_{inter,t}$
- Worst-case execution time of *t* without interference:
  - $TL_{wcet,t} = \left\lceil \frac{C(t)}{SI} \right\rceil \times SI$
- Worst-case interference from other tasks:
  - $TL_{inter,t} = TL^b_{inter,t} + TL^a_{inter,t}$
- Worst-case interference before the first scheduling interval:
  - $TL^b_{inter,t} = \begin{cases} prio(t) \times SI, & if\ first\ task \\ prio(t) - prio(pred(t)) \times SI, & if\ local\ input \\ (K-1), & else \end{cases}$
- Worst-case interference after the first scheduling interval:
  - $TL^a_{inter,t} = \left(\left\lceil \frac{C(t)}{SI} - 1 \right\rceil \right) \times SI$

Design Space Exploration (DSE) [CODES´14]

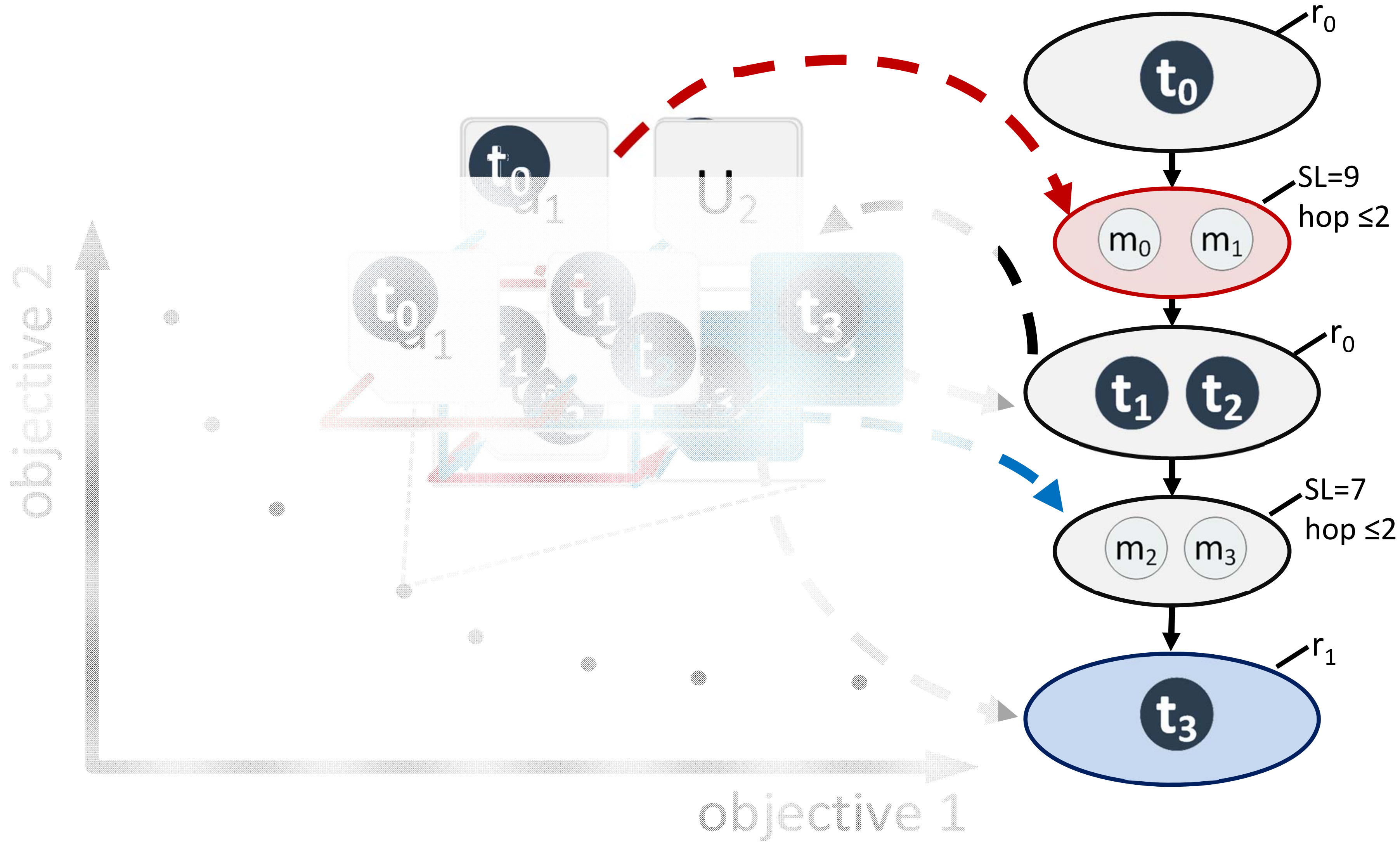
- Only mappings which do not violate the appl. deadline of concern
- Multi objective optimization

$$\max_{\forall path \in paths} \left\{ L_{path}(path, \beta, \rho) \right\} < \delta_{App}$$

- Optimization objectives maximized:
  - Average hop distance
  - Minimal hop distance
- Optimization objectives minimized:
  - Number of allocated tiles per tile type
  - Used communication resources
  - Energy

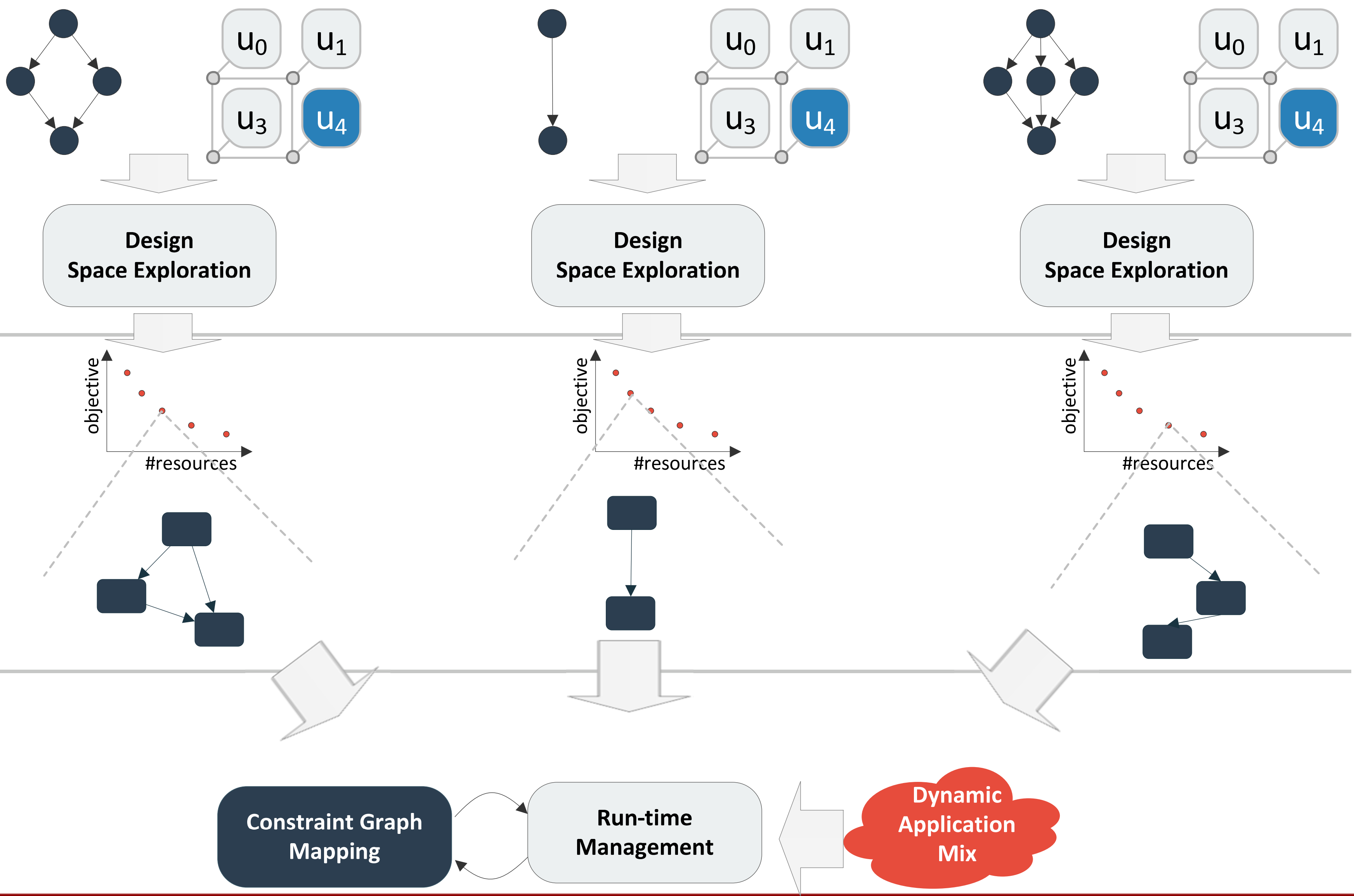


- Pareto-optimal mappings (operating points) are handed over to the run-time management system



r_0
t_0
SL=9
hop ≤2
m_0
m_1
r_0
t_1
t_2
SL=7
hop ≤2
m_2
m_3
r_1
t_3
objective 2
objective 1

Hybrid Application Mapping (HAM)

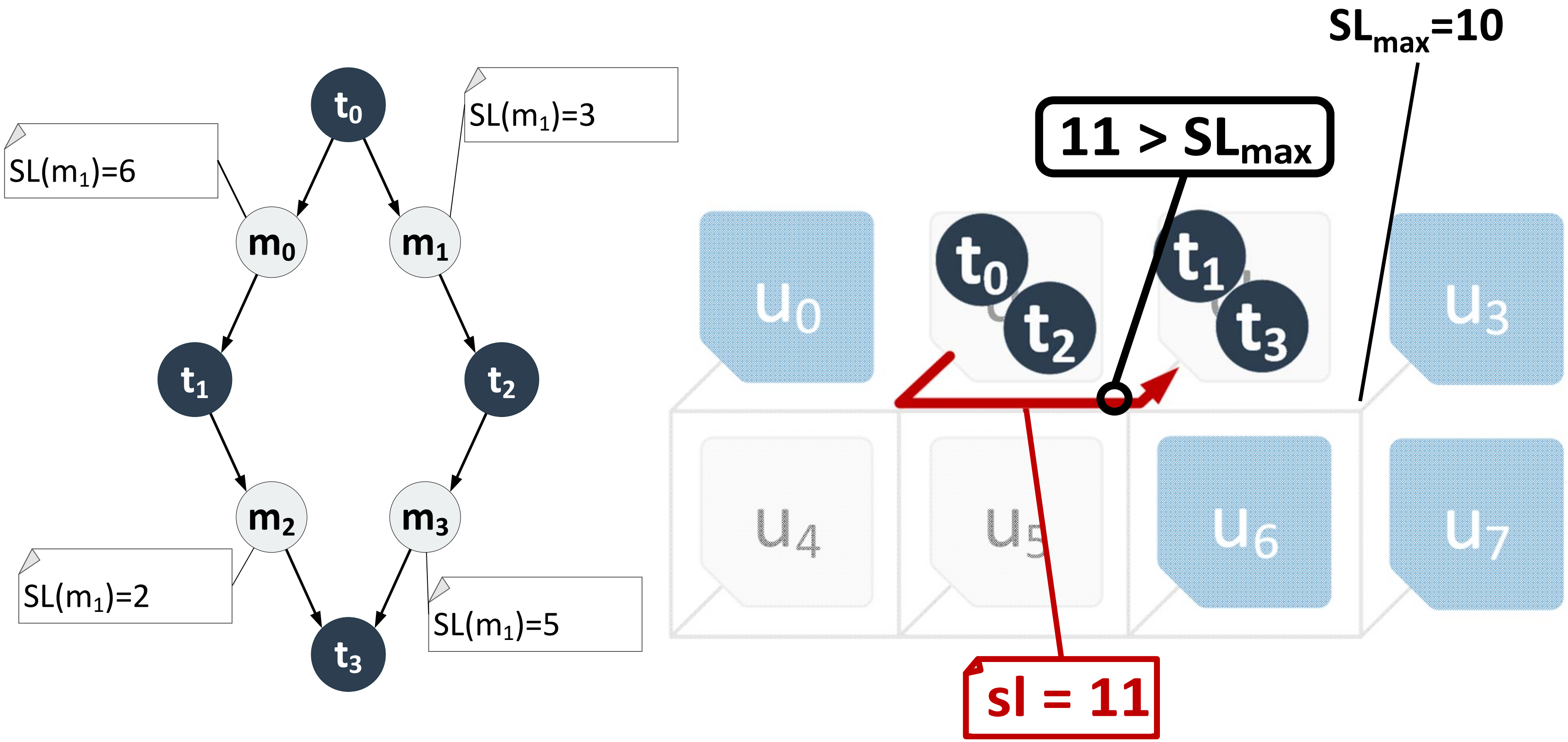


Hybrid Application Mapping (HAM)

- Scenario-based[5] and multi-mode[6] methodology optimize known application mixes during design time, but compute fixed mappings only
- Hybrid mapping approaches were introduced (e.g. [7])
  - Consider only resource availability for run-time mapping[8]
  - Communication only considered as „end-to-end latency with fixed connections“[9]

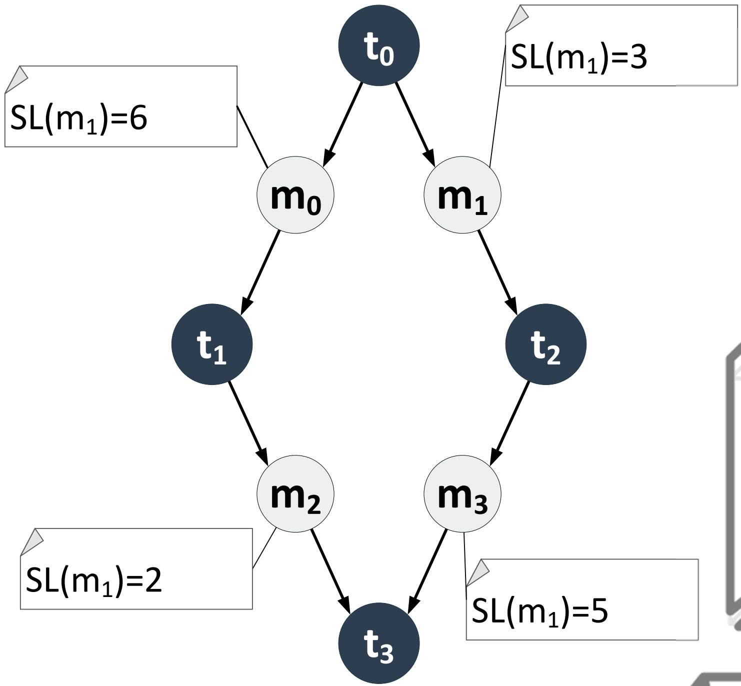
[5] P. van Stralen and A. D. Pimentel. Scenario-based design space exploration of MPSoCs. In Proceedings of Conference on Computer Design (ICCD), pp. 305–312. 2010.
[6] S. Wildermann, F. Reimann, et al. Symbolic design space exploration for multi-mode reconfigurable systems. In Proceedings of the International Conference onHardware/Software Codesign and System Synthesis (CODES+ISSS), pp. 129–138. 2011.
[7] E. Bini, G. Buttazzo, et al. Resource management on multicore systems: The ACTORS approach. Micro, IEEE, 31(3):72–81, 2011.
[8] S. Wildermann, M. Glaß, et al. Multi-objective distributed run-time resource management for many-cores. In Procceedings of Design, Automation and Test in Europe Conference and Exhibition (DATE), pp. 1–6. 2014.
[9] A. K. Singh, A. Kumar, et al. Accelerating throughput-aware runtime mapping for heterogeneous MPSoCs. ACM Transactions on Design Automation of Electronic Systems TODAES, 18(1):9:1–9:29, 2013.

Example

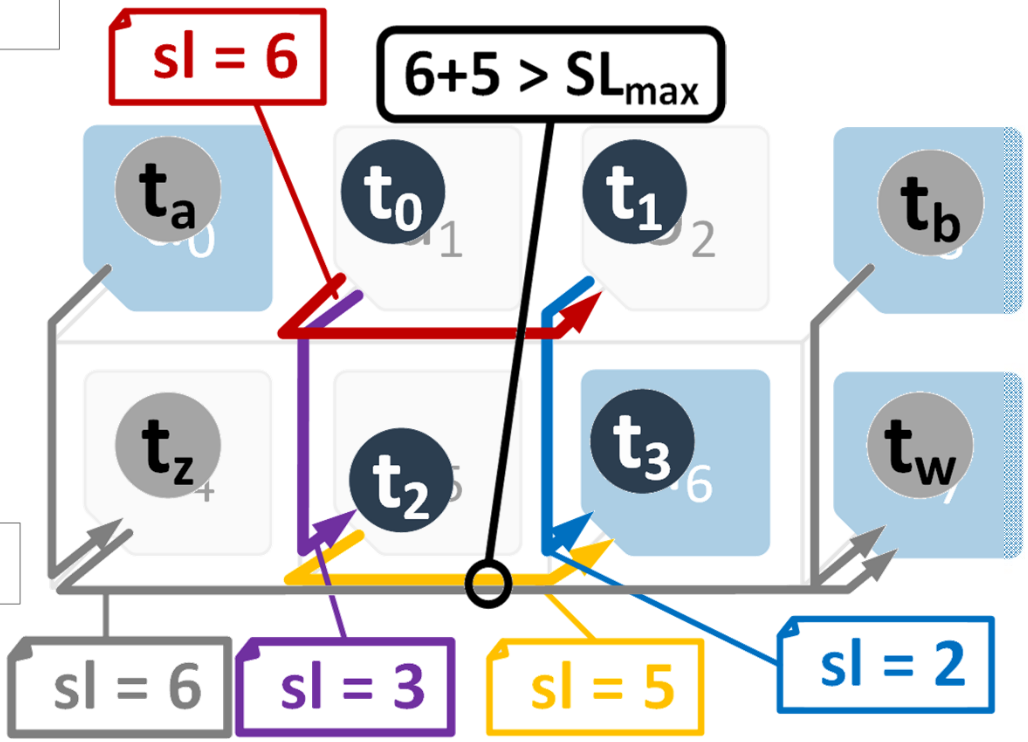


t_0
SL(m_1)=3
SL(m_1)=6
m_0
m_1
t_1
t_2
m_2
m_3
SL(m_1)=2
SL(m_1)=5
t_3
SL_{max}=10
11 > SL_{max}
u_0
t_0
t_2
t_1
t_3
u_3
u_4
u_5
u_6
u_7
sl = 11

Example

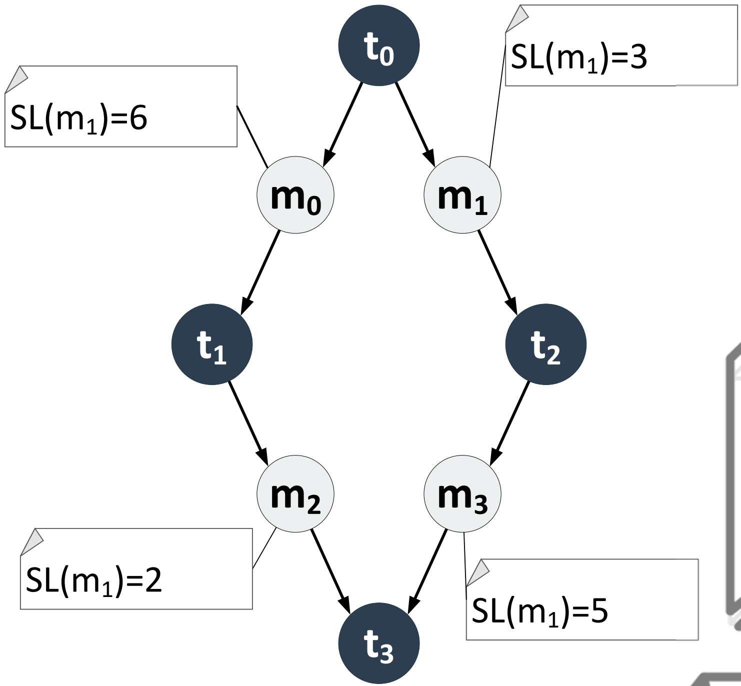


t0
SL(m1)=6
SL(m1)=3
m0
m1
t1
t2
m2
m3
SL(m1)=2
SL(m1)=5
t3

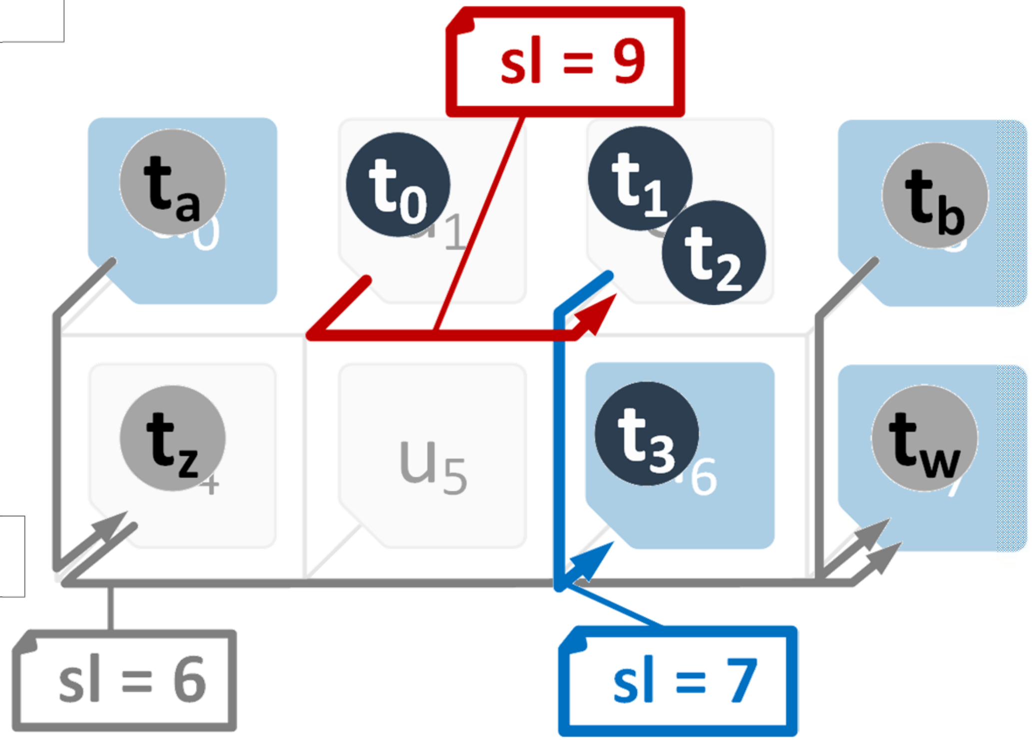


sl = 6
6+5 > SLmax
ta
t0
t1
tb
tz
t2
t3
tw
sl = 6
sl = 3
sl = 5
sl = 2

Example

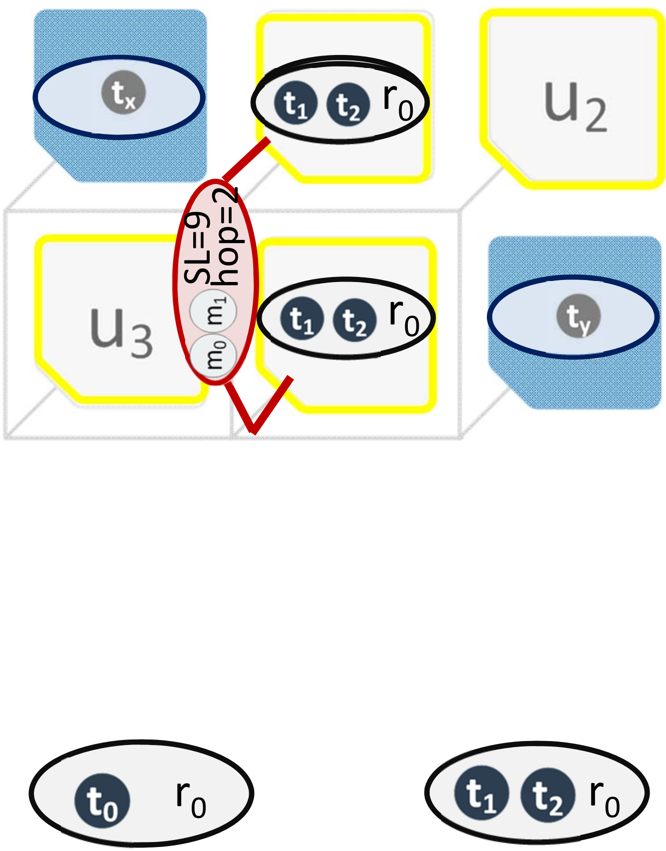


SL(m1)=6
SL(m1)=3
t0
m0
m1
t1
t2
m2
m3
t3
SL(m1)=2
SL(m1)=5



sl = 9
ta
t0
t1
t2
tb
tz
u5
t3
tw
sl = 6
sl = 7

- Backtracking algorithm for solving *constraint satisfaction problem (CSP)*




```
backtrack(A, G_C, G_NoC)
if (A is complete) then
    return A;
c = selectNextUnassignedVariable(T_C);
D_c = doForwardChecking(c, G_C, G_NoC);
for each (u ∈ D_c) do
    if u enables feasible bind. and rout. L_B then
        A' = backtrack(A ∪ ⟨c,u,L_B⟩, G_c, G_NoC);
        if (A' ≠ ∅) then
            return A';
return ∅;
```

HAM: Experiments [CODES´14]

- 15 applications taken from Embedded System Synthesis Benchmarks Suite (E3S)[4] e.g.,
  - Automotive: 18 tasks, 45 operating points
  - Consumer: 11 tasks, 79 operating points
  - Networking: 7 tasks, 48 operating points
  - Telecom: 14 tasks, 64 operating points
- DSE based on evolutionary algorithms and implemented as extension in OPT4J[7]
- Performance analysis coupled to the DSE
- Heterogeneous 6x6 NoC architecture with 3 different processor types from E3S[4]

[4] R. Dick. Embedded system synthesis benchmarks suite (E3S), 2010. http://ziyang.eecs.umich.edu/dickrp/e3s/.

[7] M. Lukasiewycz, M. Glaß, et al. Opt4J – a modular framework for meta-heuristic optimization. In Proceedings of Genetic and Evolutionary Computation Conference (GECCO), pp. 1723–1730. 2011.

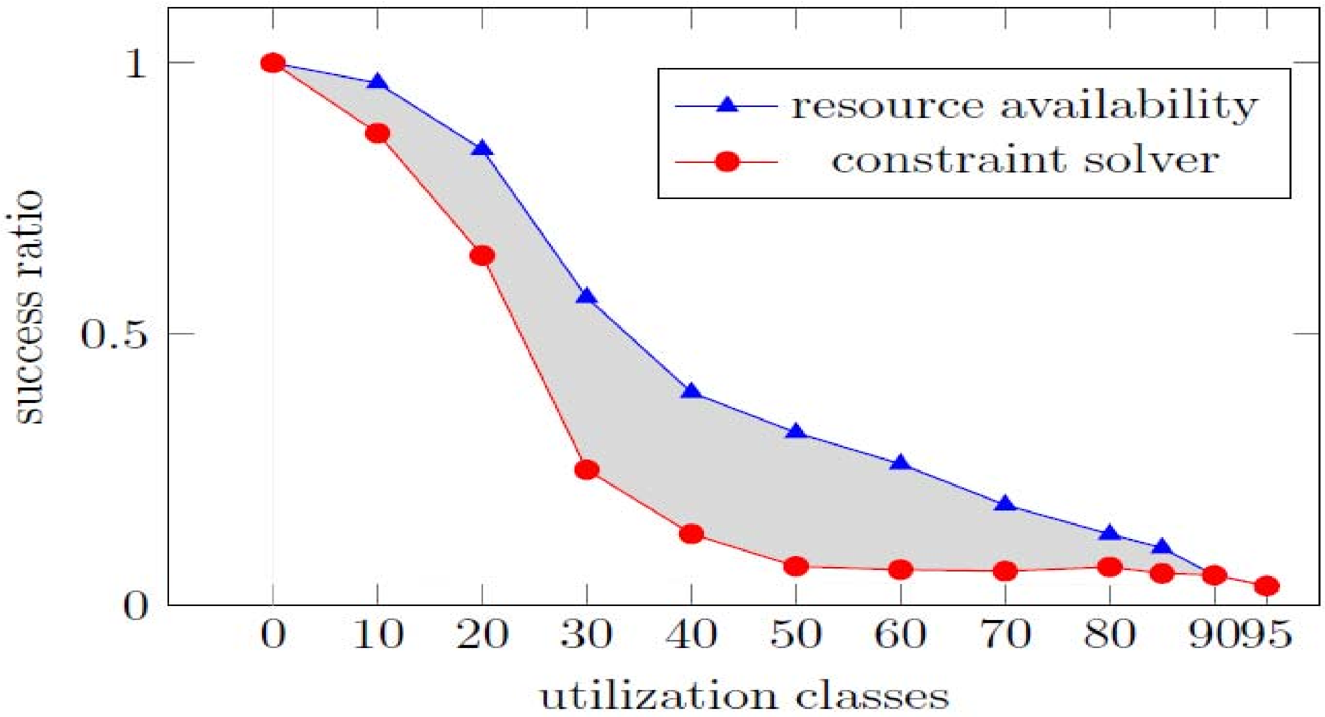
HAM: Experiments [CODES´14]

test case	#select	#operating points knap.	inc.	rep.	exec. Time [ms] knap[9]	inc.	repair
1	7	0	**5**	**6**	62.983	11	16
2	7	0	**4**	**7**	5.055	19	20
3	7	**7**	**6**	**7**	371	8	8
4	7	0	**5**	**6**	161.275	11	15
5	7	0	**5**	**6**	69.276	12	16
6	7	0	**5**	**6**	503.761	9	15
7	7	0	**5**	**7**	7.566	10	15
8	7	0	**5**	**6**	52.400	10	14
9	7	0	**4**	**7**	22.931	10	11
10	6	0	**4**	**6**	9.869	7	9

- knap: #select heuristically selected operating points (feas. acc. to resource reqs.) are tried to be mapped at once
- inc: incremental mapping of the operating points
- repair: selects other operating points if mapping fails

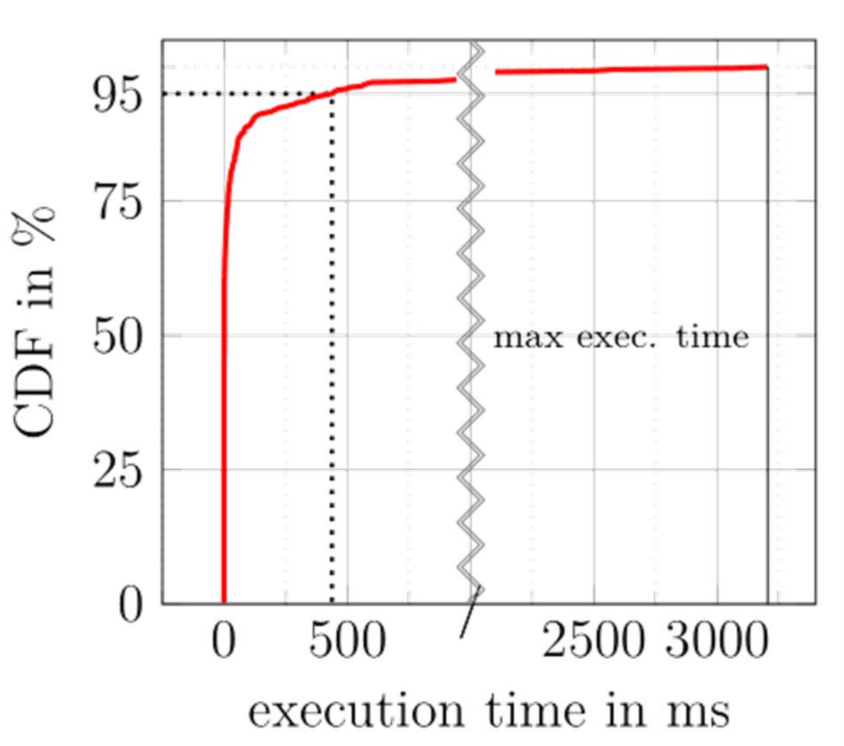
[9] S. Wildermann, M. Glaß, et al. Multi-objective distributed run-time resource management for many-cores. In Proceedings of Design, Automation and Test in Europe Conference and Exhibition (DATE), pp. 1-6. 2014.

HAM: Experiments [CODES´14]



- Considering only the resource availability for mapping may be too optimistic

HAM: Experiments [SpringerBook´18]



- 95 % of the execution times of the backtracking algorithm are within 500 ms
- To bound the execution time, a timeout can be used

Overview

Self-Embedding [NOCS’11]:
Decentralized, heuristic,
fast embedding times

Hardware
acceleration
[MOMAC’15]

Application
Mapping

Hybrid Application Mapping (HAM)
[CODES’14,SORT’15,MCSOC’16,
IT’16b,SpringerBook’18]:
Design-time analysis and run-time
mapping

Timing [CODES’14,SORT’15,
MCSOC’16,IT’16b]

Energy[CODES’14,TECS’17]

Security [SCOPES’16,IT’16a]

Reliability
[AHS’15,MOMAC’16,IT’16c]

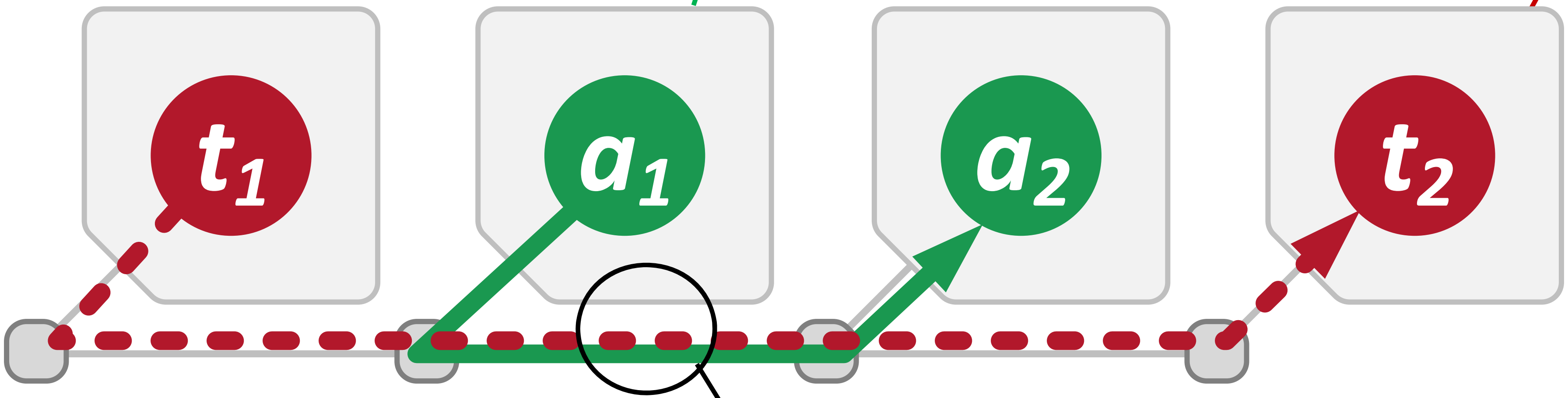
Side-Channel Attacks in NoCs

Tasks a_1 and a_2 do RSA computation:

- NoC traffic correlated to the key

Tasks t_1 and t_2 are malicious:

- Dummy traffic
- Measures time to transmit data

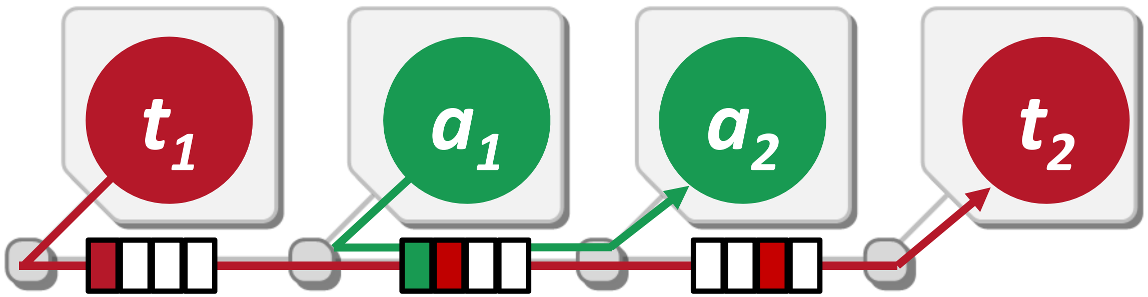


Side-channel attack in shared resource

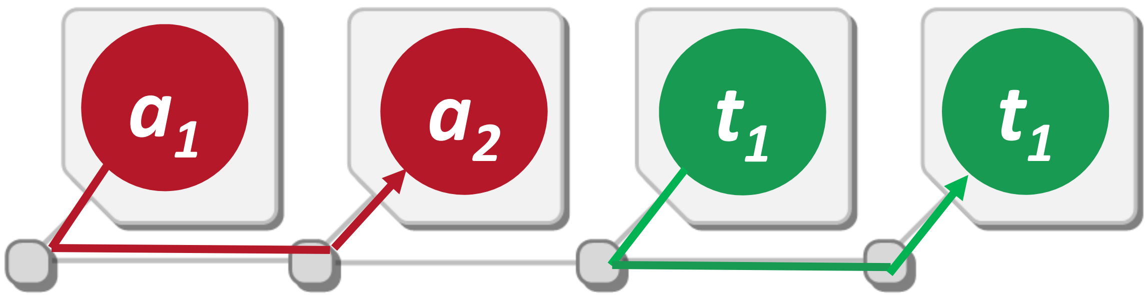
[9] Yao Wang, G. Edward Suh. "**Efficient Timing Channel Protection for On-Chip Networks.**" Proceedings of the 2012 Sixth IEEE/ACM International Symposium on Networks-on-Chip. ACM, 2012.

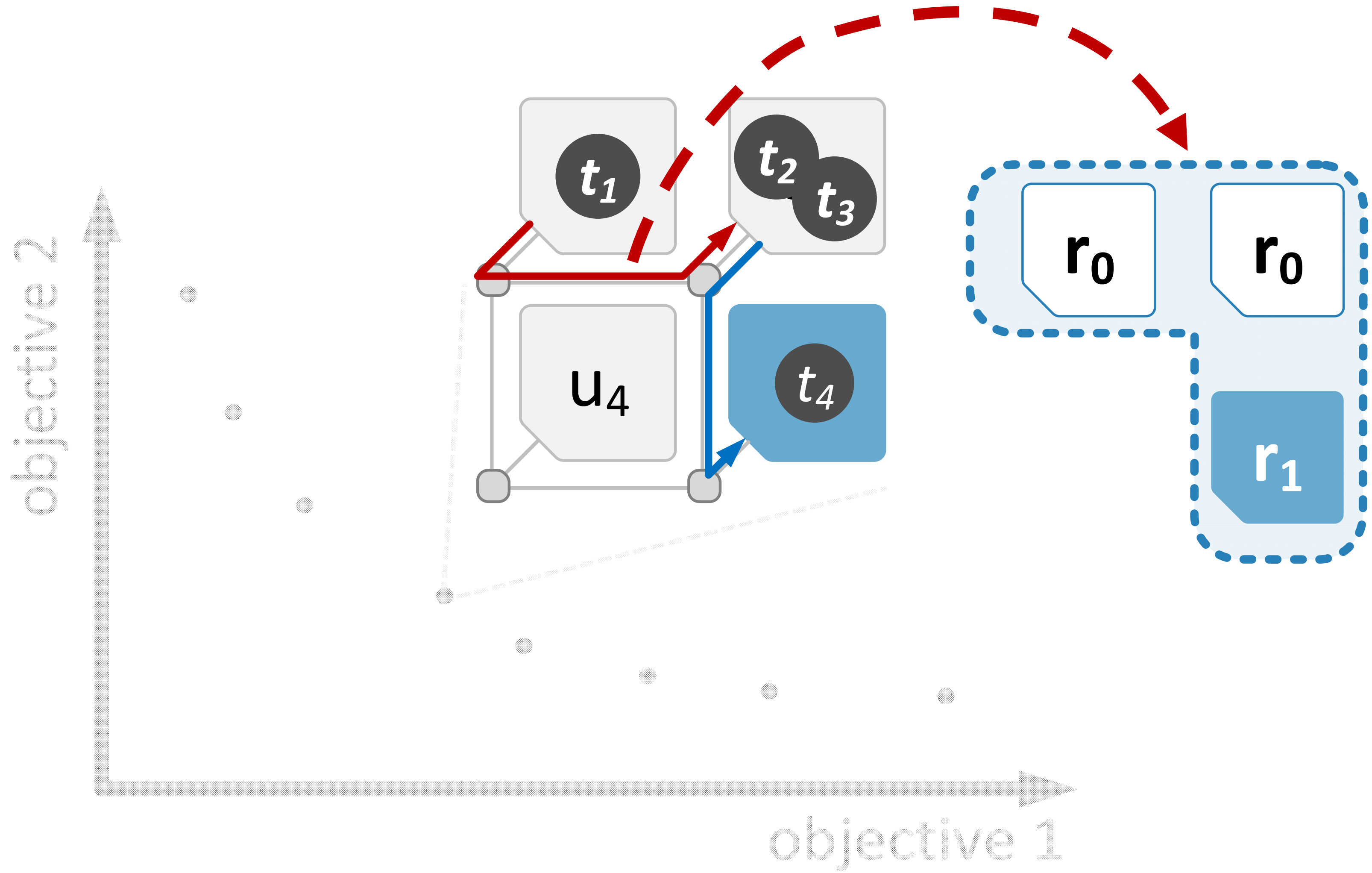
How to prevent interference and side-channel attacks?

- Strict temporal isolation (e.g., TDMA):

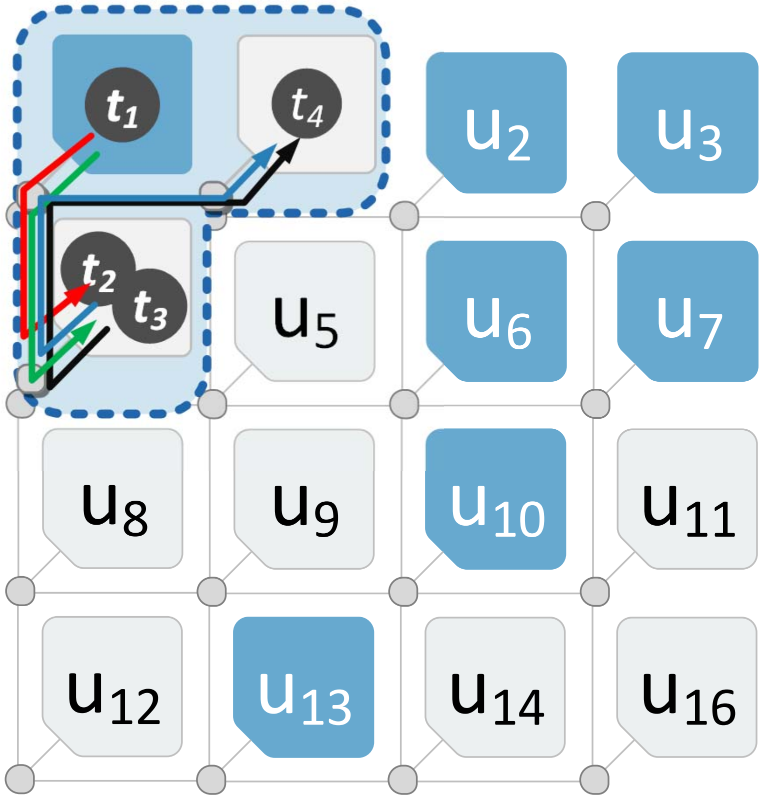


- Spatial isolation (proposed)



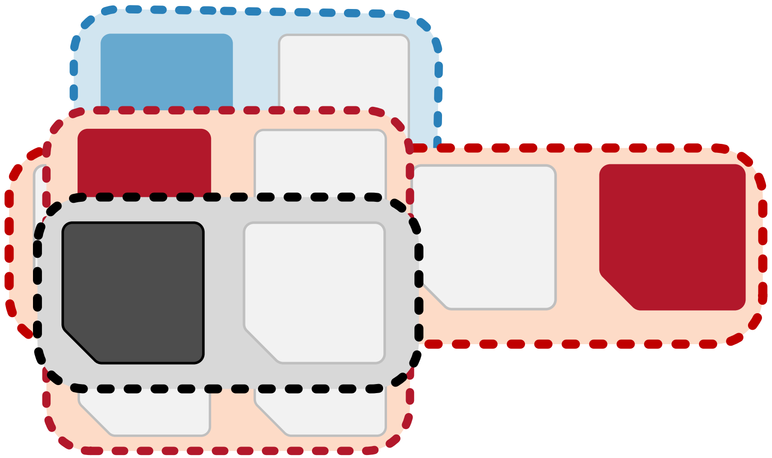


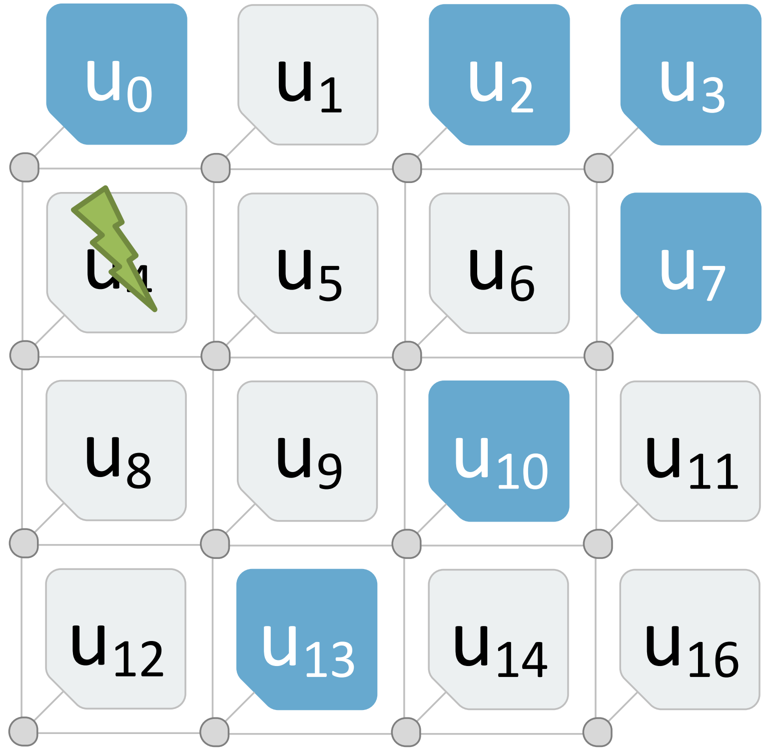
t_1
t_2
t_3
u_4
t_4
r_0
r_0
r_1
objective 2
objective 1



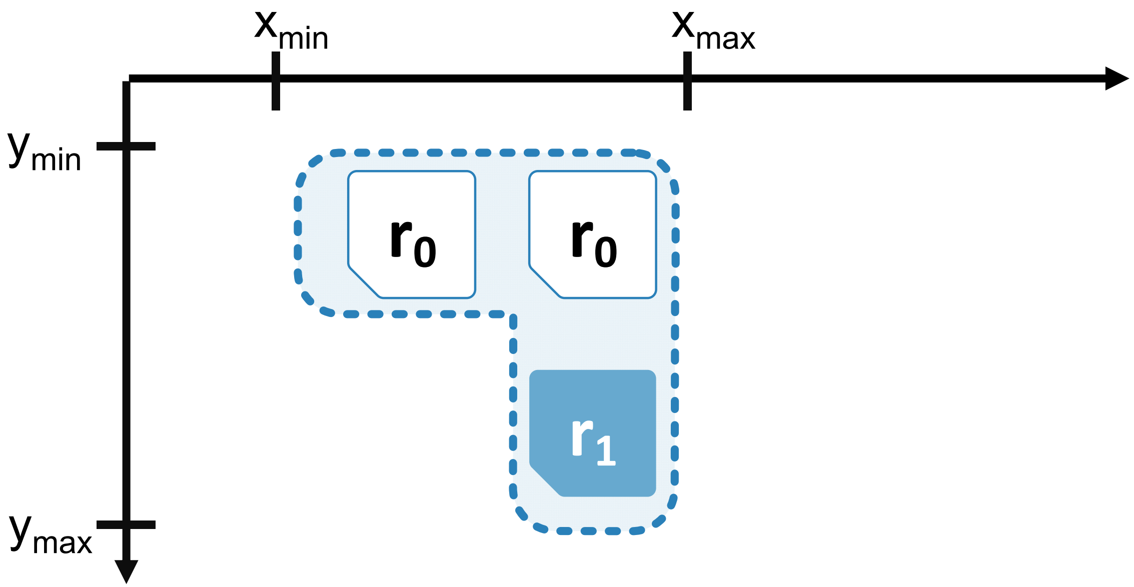
- One shape can have several *shape incarnations*
- Rotation and flipping of a shape may give equivalent mapping options

- At run time, different spatially isolated applications need to be mapped to the architecture



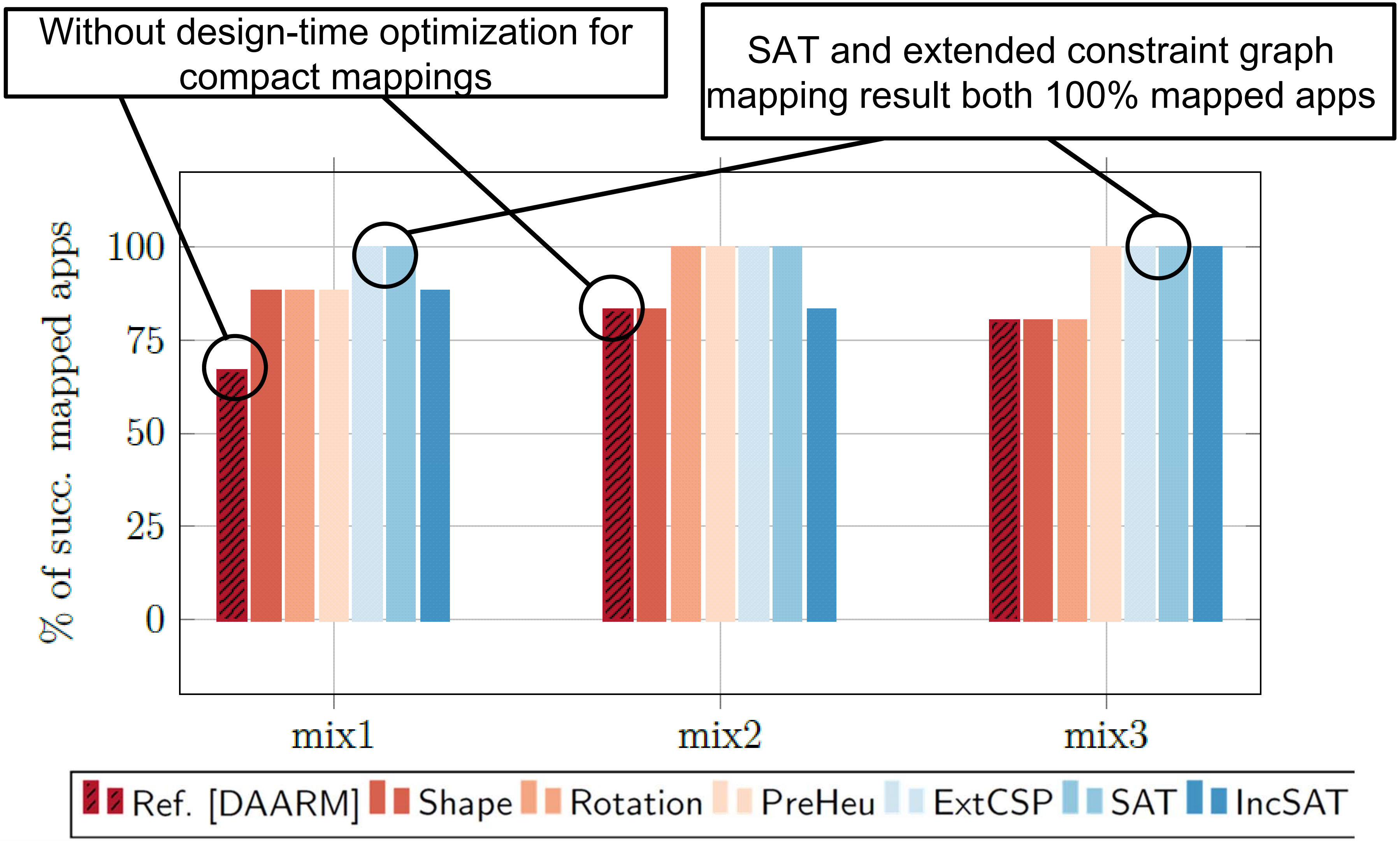


Shape-Based Design Time Optimization

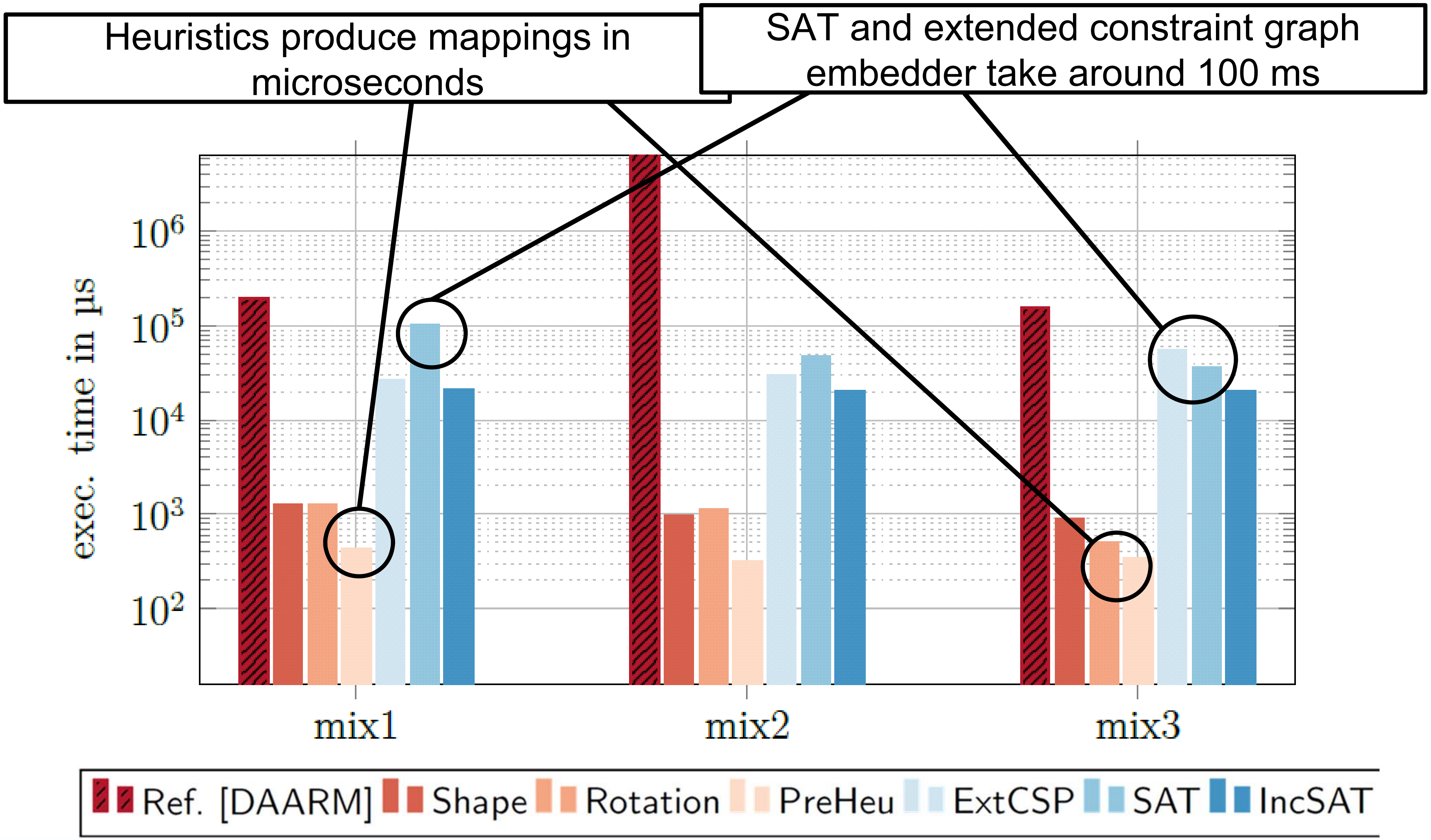


- Build convex region (depending on routing)
- Multi-Objective DSE:
  - Number of PEs: minimize ($|\#r_1|+|\#r_2|...+|holes|$)
  - Width: minimize (x_{max}-x_{min})
  - Height: minimize (y_{max}-y_{min})
  - Resources per type: minimize ($|\#r_1|$), minimize ($|\#r_2|$) …

Experimental Results [SCOPES´16]



Experimental Results [SCOPES´16]



Summary

Self-Embedding [NOCS’11]:
Decentralized, heuristic,
fast embedding times

Hardware
acceleration
[MOMAC’15]

Application
Mapping

Hybrid Application Mapping (HAM)
[CODES’14, SORT’15,MCSOC’16,
IT’16a,SpringerBook’18]:
Design-time analysis and run-time
mapping

Timing [CODES’14,SORT’15,
MCSOC’16, IT’16b]

Energy[CODES’14,TECS’17]

Security [SCOPES’16,IT’16a]

Reliability
[AHS’15,MOMAC’16,IT’16c]

Questions?

Publications

[SpriBook'18] A. Weichslgartner, S. Wildermann, M. Glaß and J. Teich. Invasive Computing for Mapping Parallel Programs to Many-Core Architectures. Springer. 2018

[IT'16a] G. Drescher, C. Erhardt, F. Freiling, J. Götzfried, D. Lohmann, P. Maene, T. Müller, I. Verbauwhede, A. Weichslgartner and S. Wildermann. Providing security on demand using invasive computing. it - Information Technology, September 30, 2016

[IT'16b] S. Wildermann, M. Bader, L. Bauer, M. Damschen, D. Gabriel, M. Gerndt, M. Glaß, J. Henkel, J. Paul, A. Pöppl, S. Roloff, T. Schwarzer, G. Snelting, W. Stechele, J. Teich, A. Weichslgartner and A. Zwinkau. Invasive Computing for Timing-Predictable Stream Processing on MPSoCs. it - Information Technology, September 30, 2016.

[IT'16c] V. Lari, A. Weichslgartner, A. Tanase, M. Witterauf, F. Khosravi, J. Teich, J. Heißwolf, S. Friederich and J. Becker. Providing Fault Tolerance Through Invasive Computing. it - Information Technology, October 19, 2016.

[MCSOC'16] J. Teich, M. Glaß, S. Roloff, W. Schröder-Preikschat, G. Snelting, A. Weichslgartner and S. Wildermann. Language and Compilation of Parallel Programs for *-Predictable MPSoC Execution using Invasive Computing. In Proceedings of the 10th IEEE International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoC-16), pp. 313-320, Lyon, France, September 21-23, 2016.

[SCOPES'16] A. Weichslgartner, S. Wildermann, J. Götzfried, F. Freiling, M. Glaß and J. Teich. Design-Time/Run-Time Mapping of Security-Critical Applications in Heterogeneous MPSoCs.

In Proceedings of the 19th International Workshop on Software and Compilers for Embedded Systems (SCOPES), pp 153-162, Sankt Goar, Germany, May 23-25, 2016

Publications

[MOMAC'15] A. Weichslgartner, J. Heisswolf, A. Zaib, T. Wild, A. Herkersdorf, J. Becker and J. Teich. Position Paper: Towards Hardware-Assisted Decentralized Mapping of Applications for Heterogeneous NoC Architectures. In Proceedings of the second International Workshop on Multi-Objective Many-Core Design (MOMAC) in conjunction with International Conference on Architecture of Computing Systems (ARCS), pp. 4, Porto, Portugal, March 24, 2015

[CODES'14] A. Weichslgartner, D. Gangadharan, S. Wildermann, M. Glaß and J. Teich. DAARM: Design-Time Application Analysis and Run-Time Mapping for Predictable Execution in Many-Core Systems . In Proceedings of the International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS 2014), pp. 34:1 – 34:10, New Delhi, India, October 12-17, 2014

[DAC'14] J. Heisswolf, A. Zaib, A. Zwinkau, S. Kobbe, A. Weichslgartner, J. Teich, J. Henkel, G. Snelting, A. Herkersdorf and J. Becker. CAP: Communication Aware Programming. In Proceedings of the 51st Design Automation Conference (DAC 2014), pp. 105:1 – 105:6, San Francisco, CA, USA, June 1-5, 2014.

[MOMAC'14] J. Heisswolf, A. Zaib, A. Weichslgartner, M. Karle, M. Singh, T. Wild, J. Teich, A. Herkersdorf and J. Becker. The Invasive Network on Chip - A Multi-Objective Many-Core Communication Infrastructure. In Proceedings of the first International Workshop on Multi-Objective Many-Core Design (MOMAC) in conjunction with International Conference on Architecture of Computing Systems (ARCS), pp. 1-8, Lübeck, Germany, Feb. 25, 2014.

Publications

[DSD'13] A. Zaib, J. Heisswolf, A. Weichslgartner, T. Wild, J. Teich, J. Becker and A. Herkersdorf. AUTO-GS: Self-optimization of NoC Traffic Through Hardware Managed Virtual Connections. In Proceedings of the 16th Euromicro Conference on Digital System Design (DSD), pp. 761–768, Santander, Spain, Sep. 4-6, 2013.

[TRETS'13] J. Heisswolf, A. Zaib, A. Weichslgartner, T. Wild, J. Teich, A. Herkersdorf and J. Becker. Virtual Networks - Distributed Communication Resource Management. In ACM Transactions on Reconfigurable Technology and Systems (TRETS), 6(2):8:1–8:14, Aug., 2013.

[SCOPES'13] S. Roloff, A. Weichslgartner, J. Heißwolf, F. Hannig and J. Teich. NoC Simulation in Heterogeneous Architectures for PGAS Programming Model. In Proceedings of the 16th International Workshop on Software and Compilers for Embedded Systems (M-SCOPES), pp. 77-85, St. Goar, Germany, Jun. 19-21, 2013.

[RAW'13] J. Heisswolf, A. Weichslgartner, A. Zaib, R. König, T. Wild, J. Teich, A. Herkersdorf and J. Becker. Hardware Supported Adaptive Data Collection for Networks on Chip. Proceedings of 20th Reconfigurable Architectures Workshop (RAW 2013), Boston, USA, pp. 153–162, May 2013.

[FDL'12] J. Teich, A. Weichslgartner, B. Oechslein and W. Schröder-Preikschat. Invasive Computing - Concepts and Overheads. In Proceedings of the Forum on Specification & Design Languages (FDL), pp. 193-200, Vienna, Austria, September 18-20, 2012.

Publications

[RAW'12] J. Heisswolf, A. Zaib, A. Weichslgartner, R. König, T. Wild, J. Teich, A. Herkersdorf and J. Becker. Hardware-assisted Decentralized Resource Management for Networks on Chip with QoS. In Proceedings of 19th Reconfigurable Architectures Workshop (RAW 2012), pp. 234 - 241, Shanghai, China, May 2012.

[FPL'11] D. Ziener, S. Wildermann, A. Oetken, A. Weichslgartner and J. Teich. A Flexible Smart Camera System based on a Partially Reconfigurable Dynamic FPGA-SoC. In Proceedings of the Workshop on Computer Vision on Low-Power Reconfigurable Architectures at FPL 2011, pp. 29-30, Chania, Crete, Greece, September 4, 2011.

[NOCS'11] A. Weichslgartner, S. Wildermann and J. Teich. Dynamic Decentralized Mapping of Tree-Structured Applications on NoC Architectures. In Proceedings of the Fifth ACM/IEEE International Symposium on Networks-on-Chip (NOCS 2011), pp. 201-208, Pittsburgh, USA, May 1-4, 2011.