Influence of different system abstractions on the performance analysis of distributed real-time systems

TRESOR seminar, EPFL Lausanne

29. March 2007

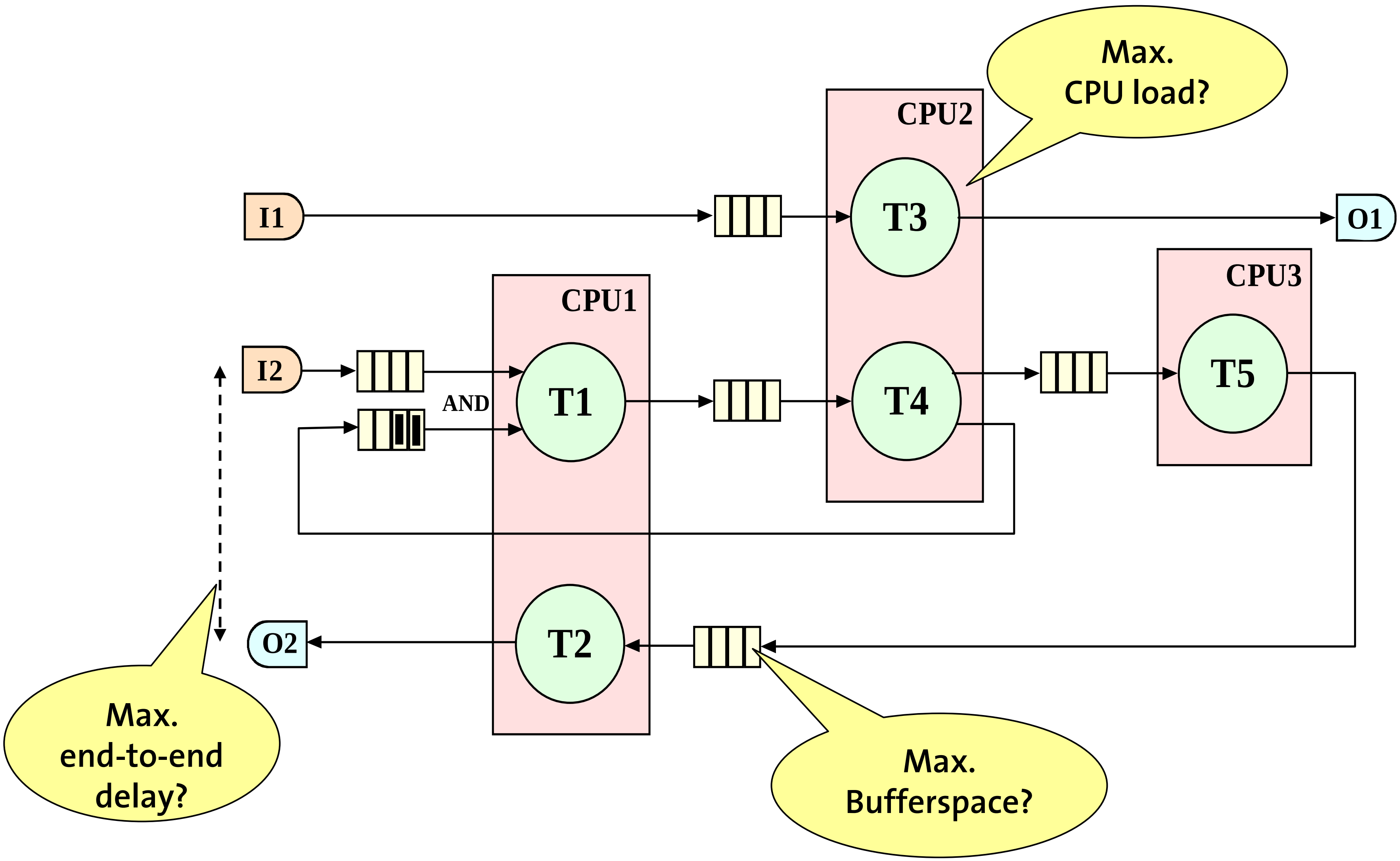
Simon Perathoner, Ernesto Wandeler, Lothar Thiele

Computer Engineering and Networks Laboratory
ETH Zürich, Switzerland

Outline

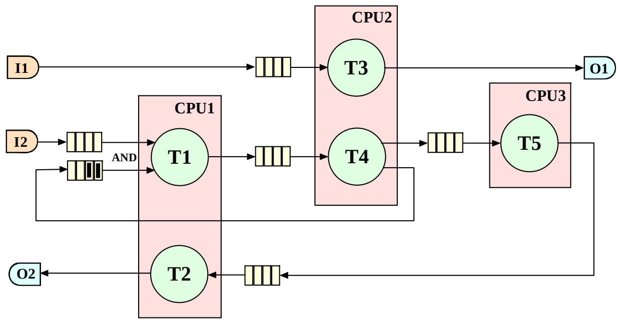
- Motivation
- Abstractions
- Benchmarks
- Conclusions

System level performance analysis



Formal analysis methods

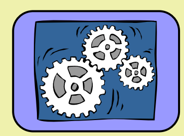
Distributed system



Abstraction 3

$$r_i = C_i + \sum_{\forall j \in hp(i)} \lceil \frac{r_i}{T_j} \rceil \, C_j$$

Analysis method 3



Performance values

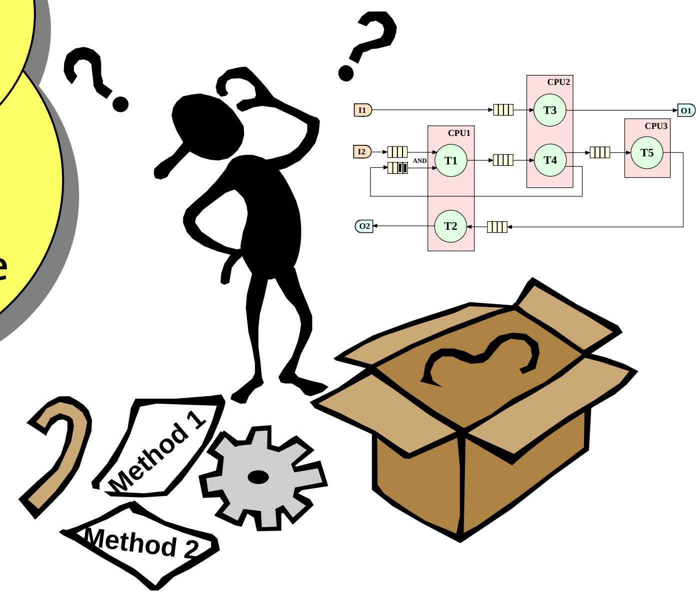
Motivating questions

- What is the influence of the different models on the analysis accuracy ?
- Does abstraction matter ?
- Which abstraction is best suited for a given system ?

Evaluation and comparison of abstractions is needed !

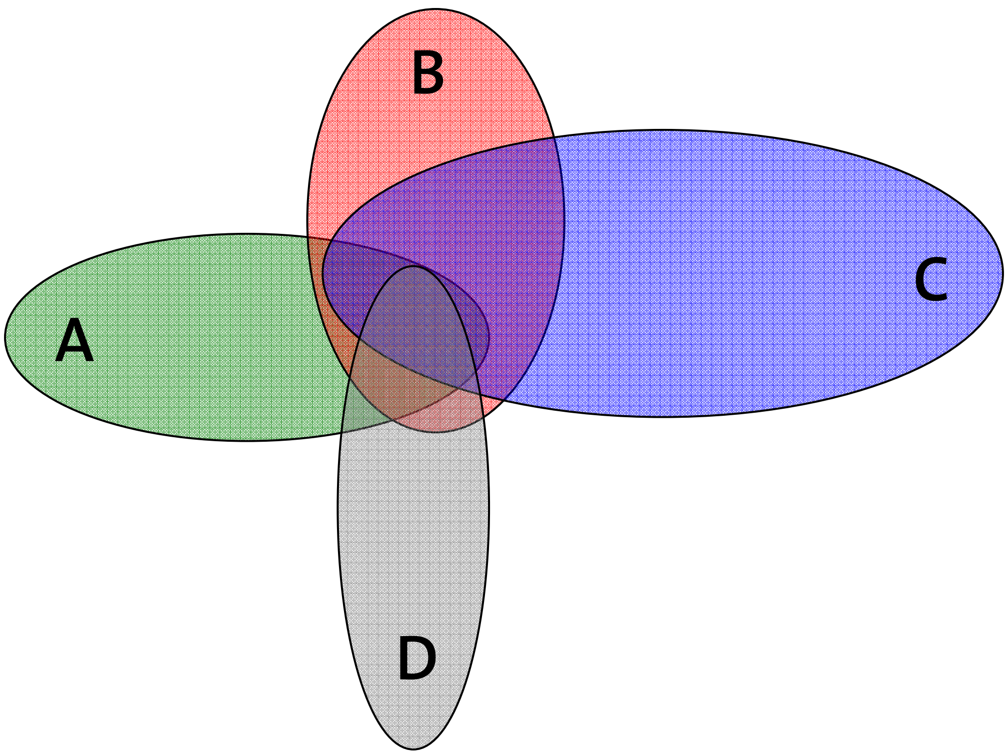
How can we compare different abstractions ?

modular
performance metrics
modeling effort
correctness
scope
scalability
usability
analysis time
accuracy
exact
efficient implementation
simple
compositional
holistic
stepwise refinement
tool support
learning curve
easy to use



What makes a direct comparison difficult?

- Many aspects can not be quantified
- Models cover different scenarios:



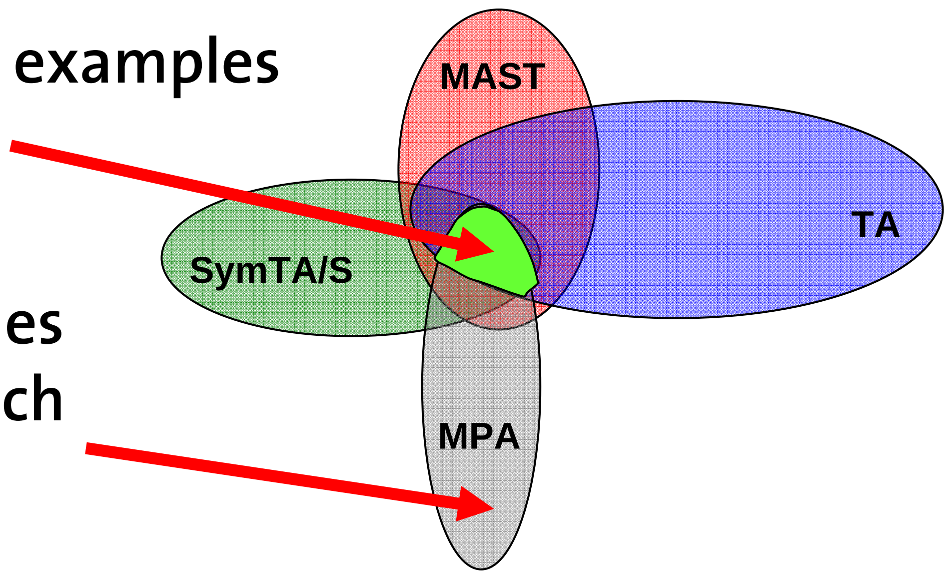
Intention

Compare models and methods that analyze the timing properties of distributed systems:

- SymTA/S [Richter *et al.*]
- MPA-RTC [Thiele *et al.*]
- MAST [González Harbour *et al.*]
- Timed automata based analysis [Yi *et al.*]
- …

Approach

- Leiden Workshop on Distributed Embedded Systems: http://www.tik.ee.ethz.ch/~leiden05/
- Define a set of benchmark examples that cover common area
- Define benchmark examples that show the power of each method



Expected (long term) results

- Understand the modeling power of different methods
- Understand the relation between models and analysis accuracy
- Improve methods by combining ideas and abstractions

Contributions

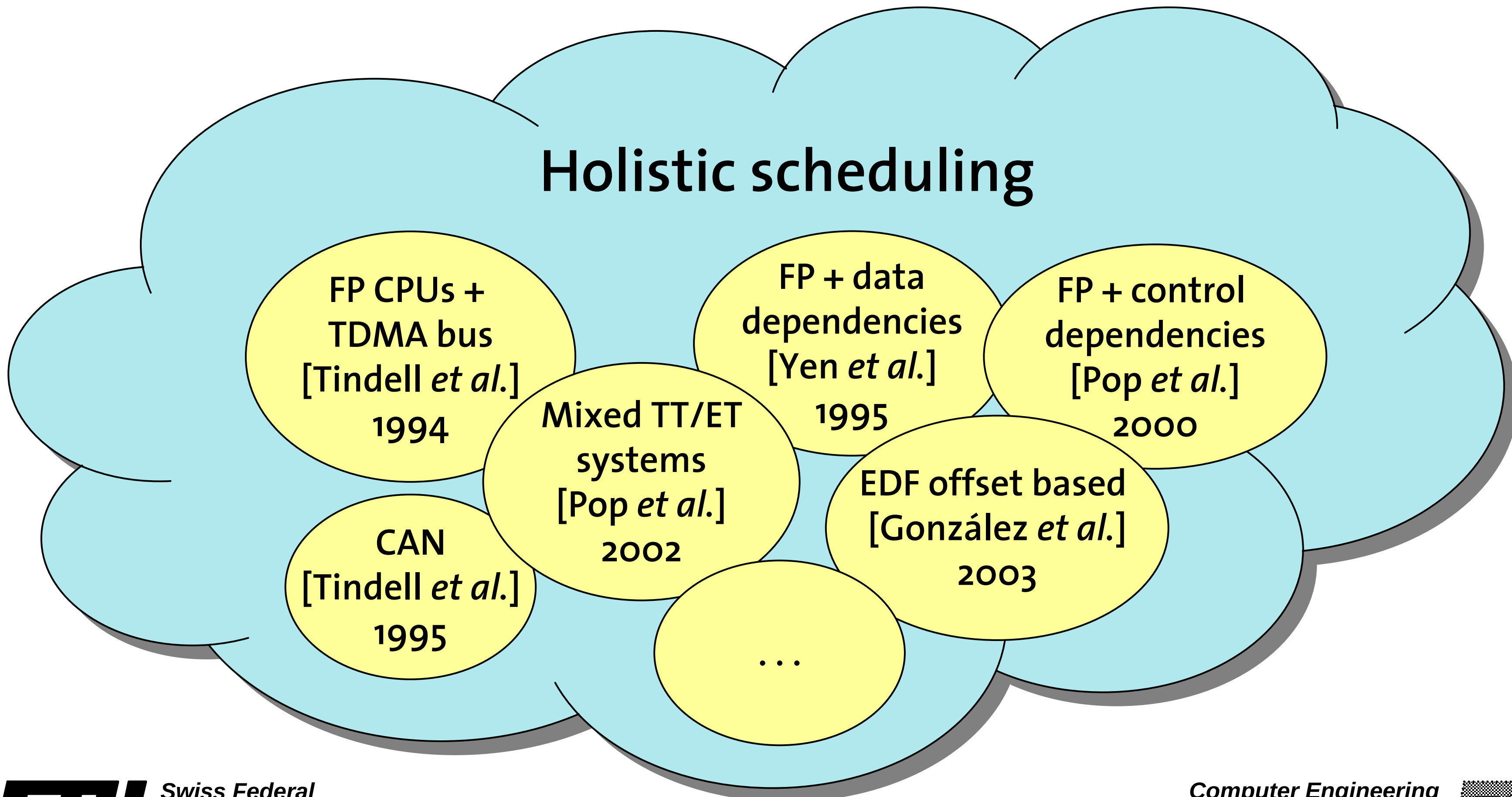
- We define a set of benchmark systems aimed at the evaluation of performance analysis techniques
- We apply different analysis methods to the benchmark systems and compare the results obtained in terms of accuracy and analysis times
- We point out several analysis difficulties and investigate the causes for deviating results

Outline

- Motivation
- Abstractions
- Benchmarks
- Conclusions

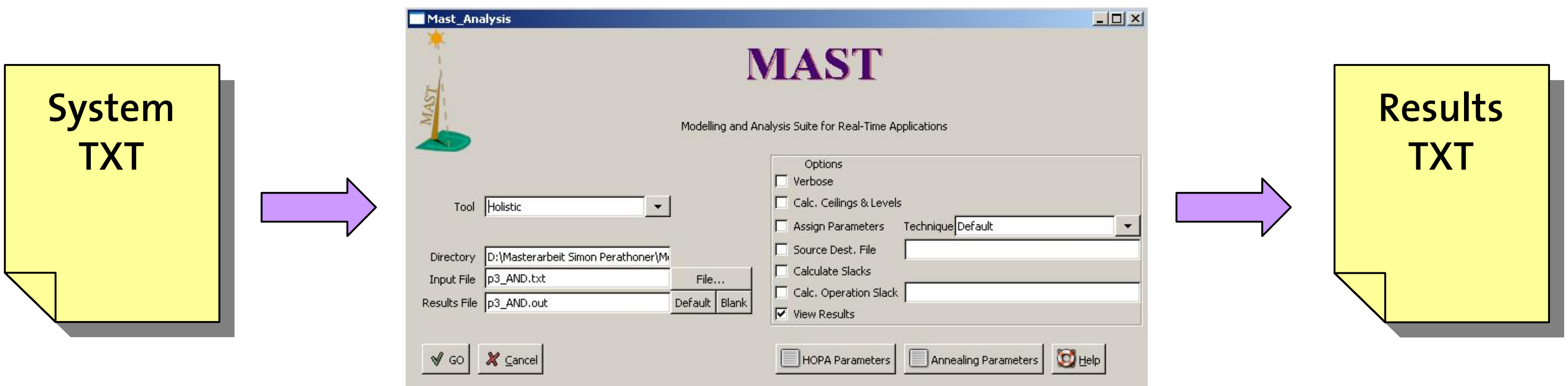
Abstraction 1 - Holistic scheduling

Basic concept: extend concepts of classical scheduling theory to distributed systems



Holistic scheduling – MAST tool

MAST - The Modeling and Analysis Suite for Real-Time Applications [González Harbour *et al.*]



Abstrction 2 – The SymTA/S approach

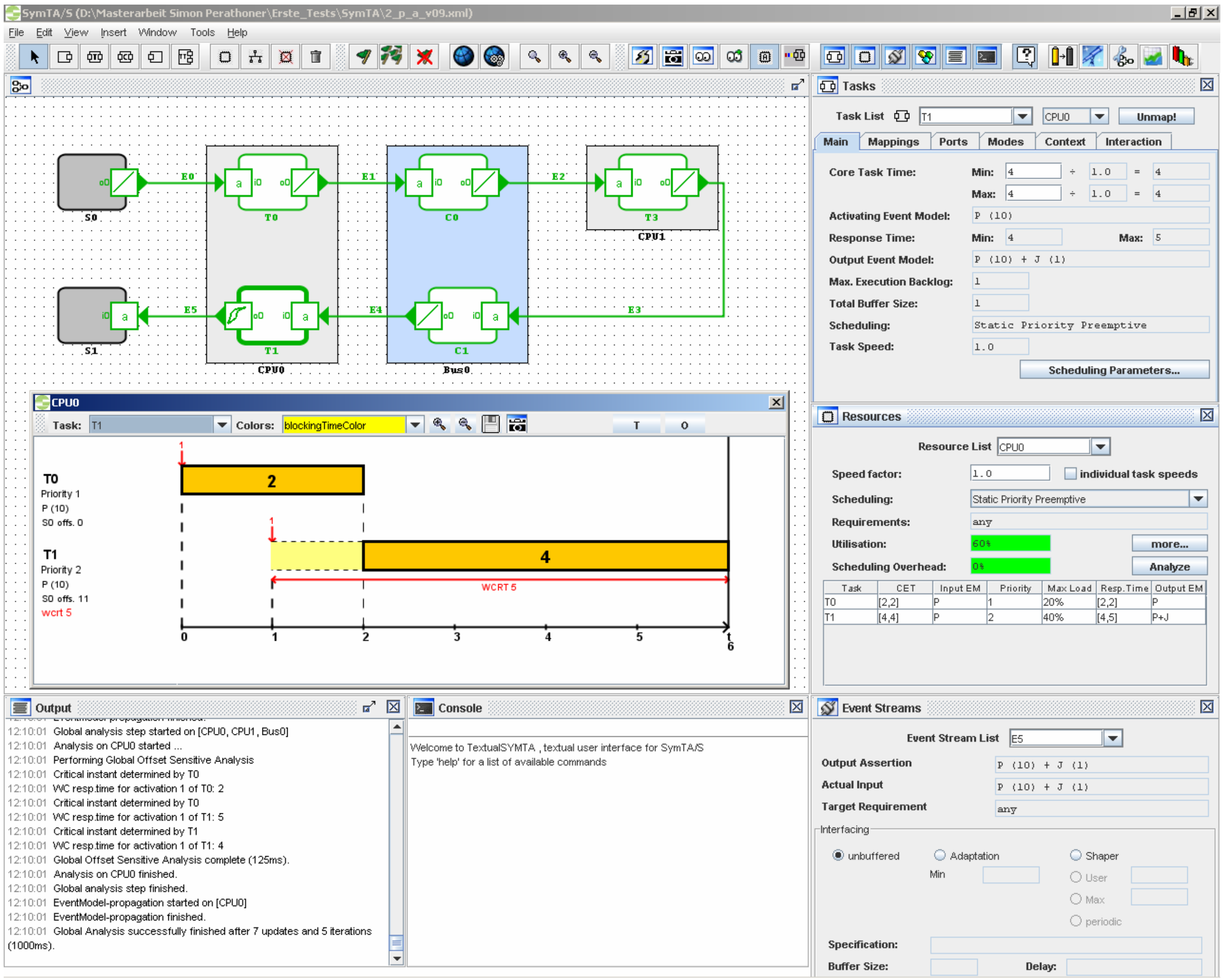
Basic concept: Application of classical scheduling techniques at resource level and propagation of results to next component

Problem: The local analysis techniques require the input event streams to fit given standard event models

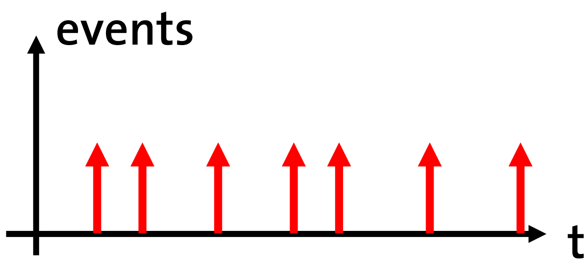
Solution: Use appropriate interfaces: EMIFs & EAFs

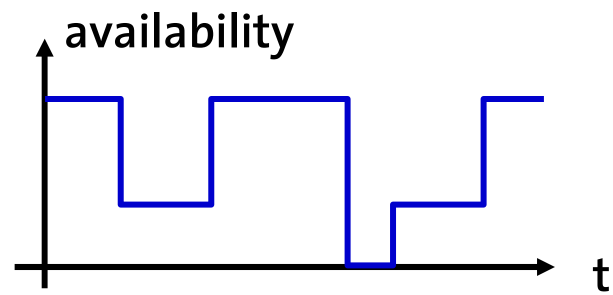
SymTA/S – Tool

SYMTA VISION



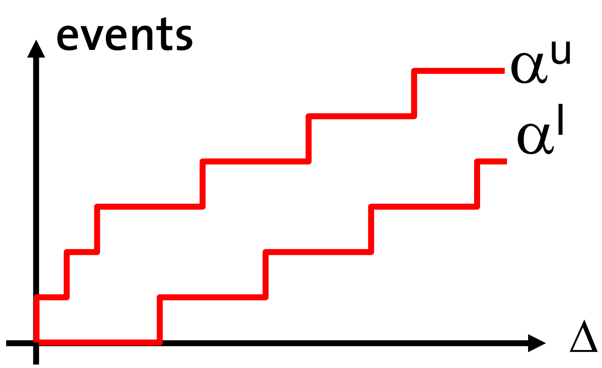
Abstraction 3 – MPA-RTC



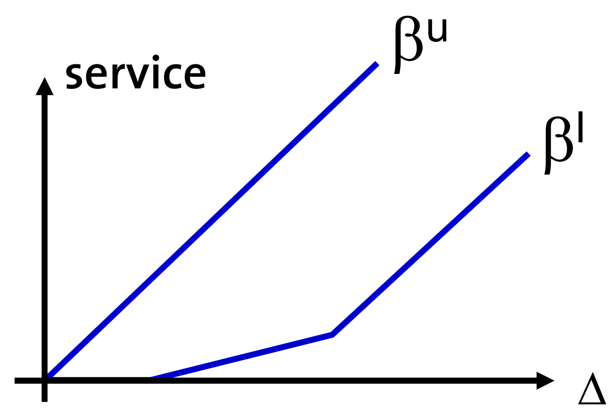


Load model

Service model

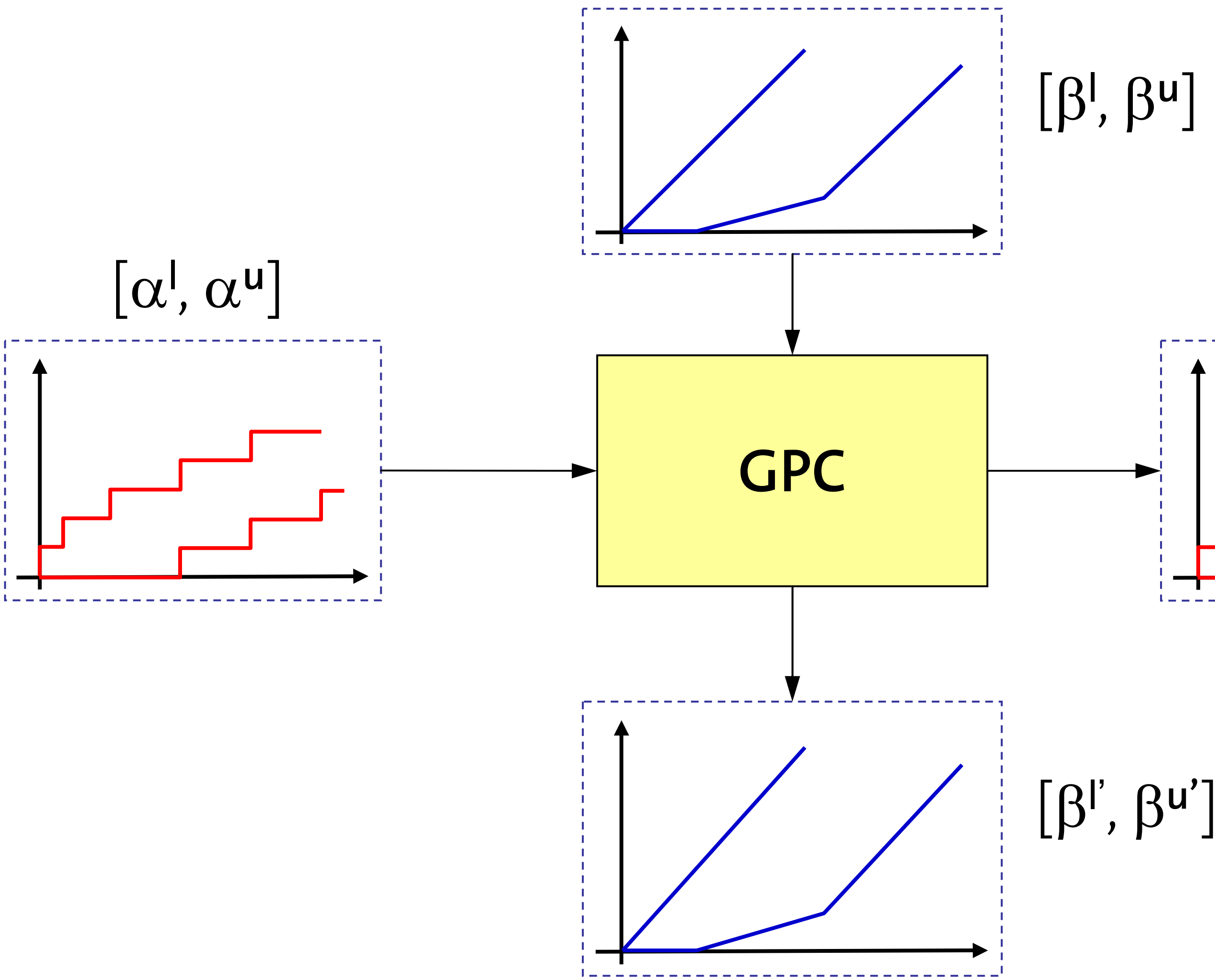


Arrival curves

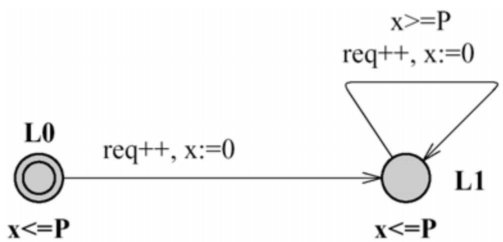


Service curves

Abstraction 3 – MPA-RTC



Abstraction 4 - TA based performance analysis

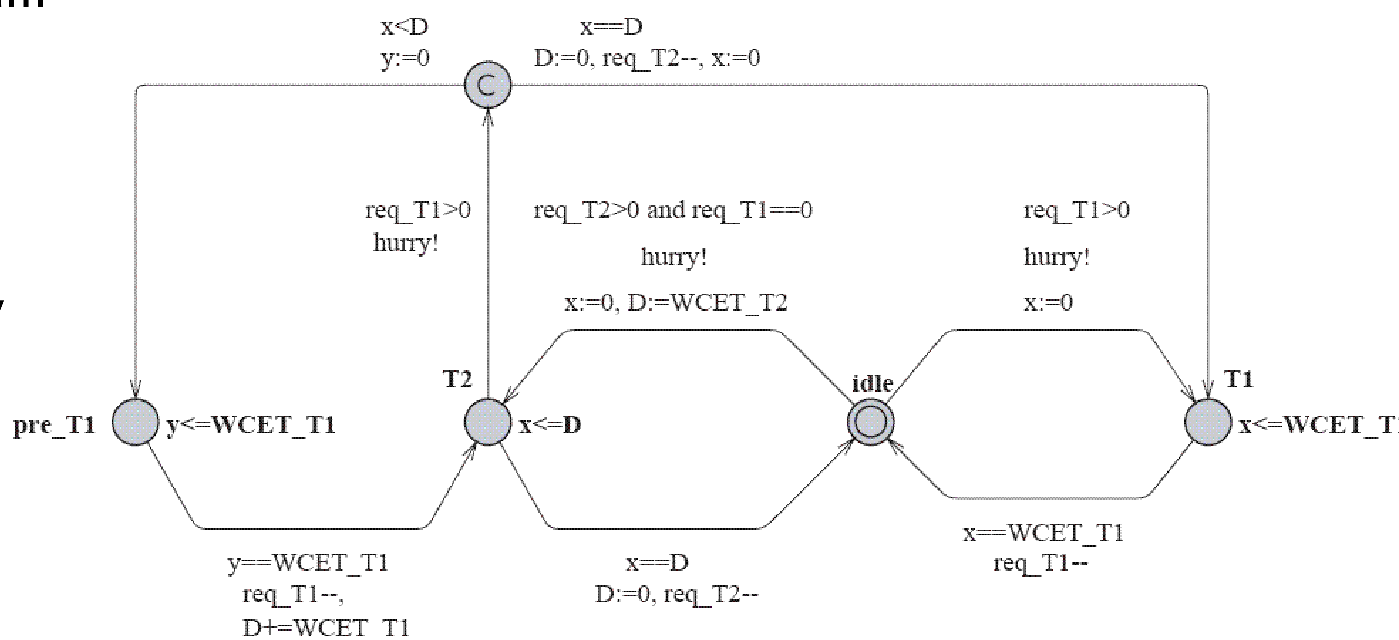


periodic stream

Verification of performance properties by model checking (UPPAAL)

Exact performance values

fixed priority scheduling



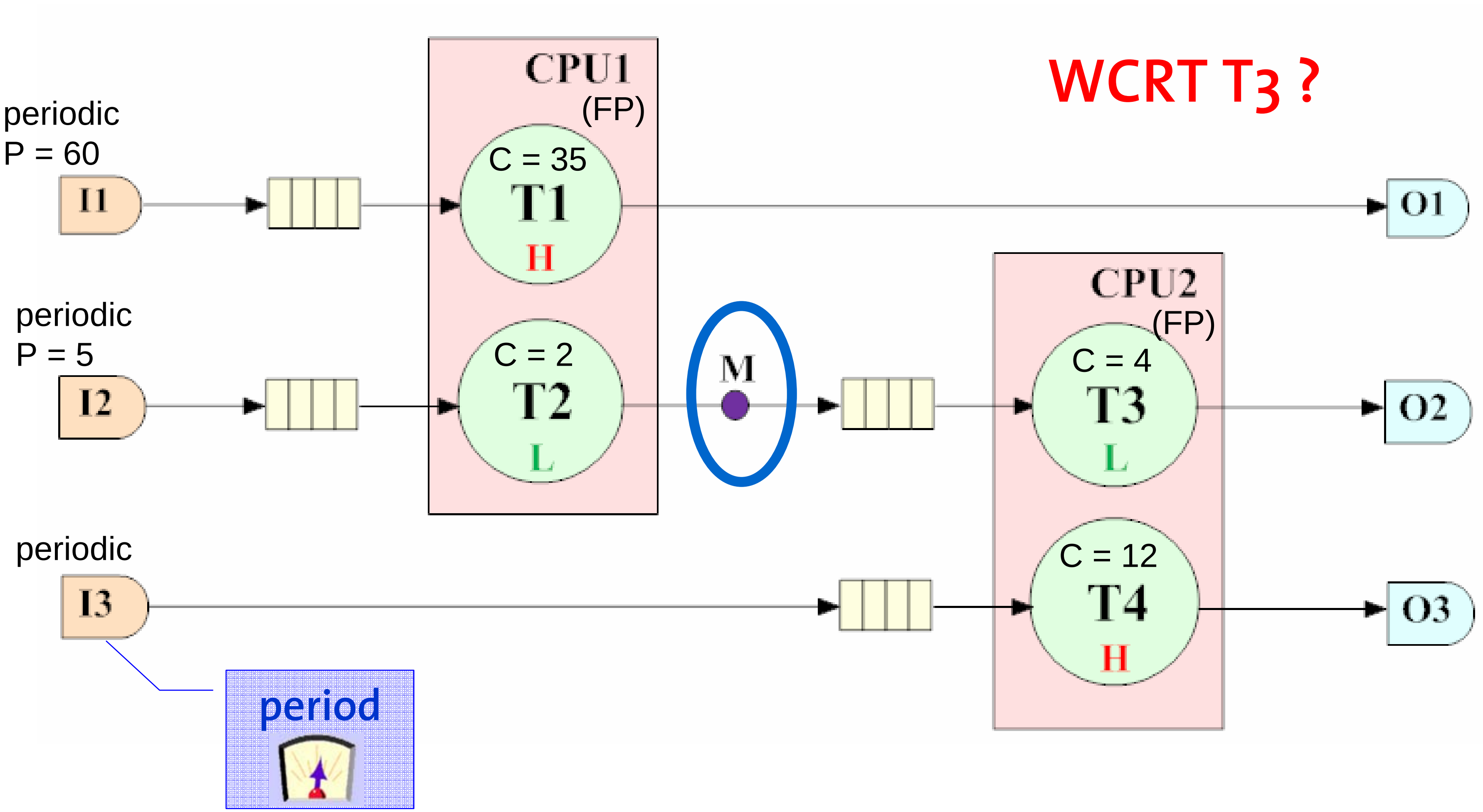
Outline

- Motivation
- Abstractions
- Benchmarks
- Conclusions

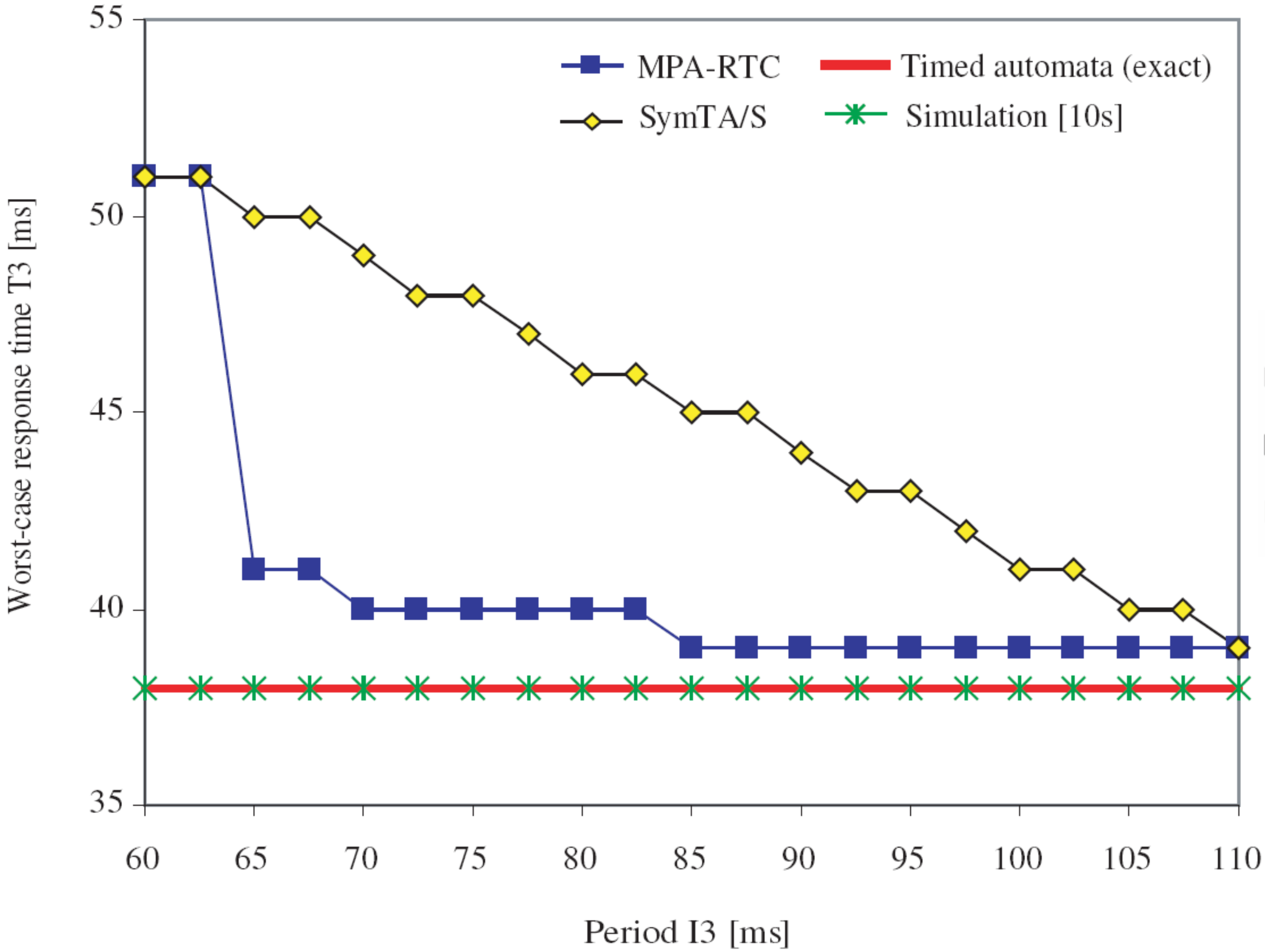
Benchmarks

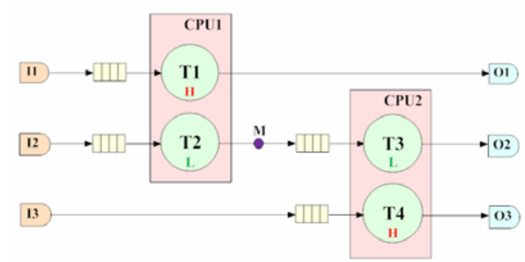
- Pay burst only once
- Complex activation pattern
- Variable feedback
- Cyclic dependencies
- AND/OR task activation
- Intra-context information
- Workload correlation
- Data dependencies

Benchmark 1 – Complex activation pattern



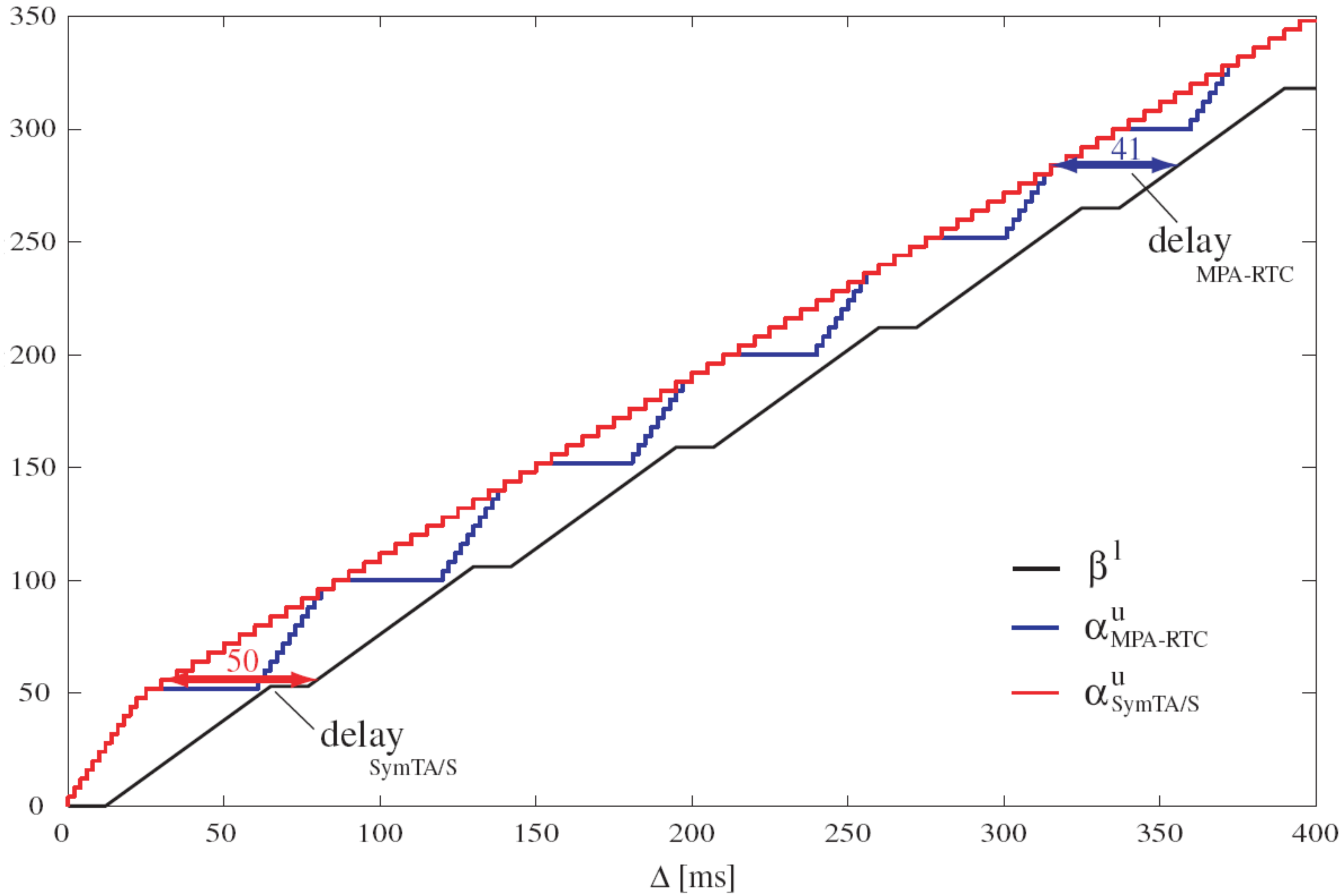
Benchmark 1 – Analysis results



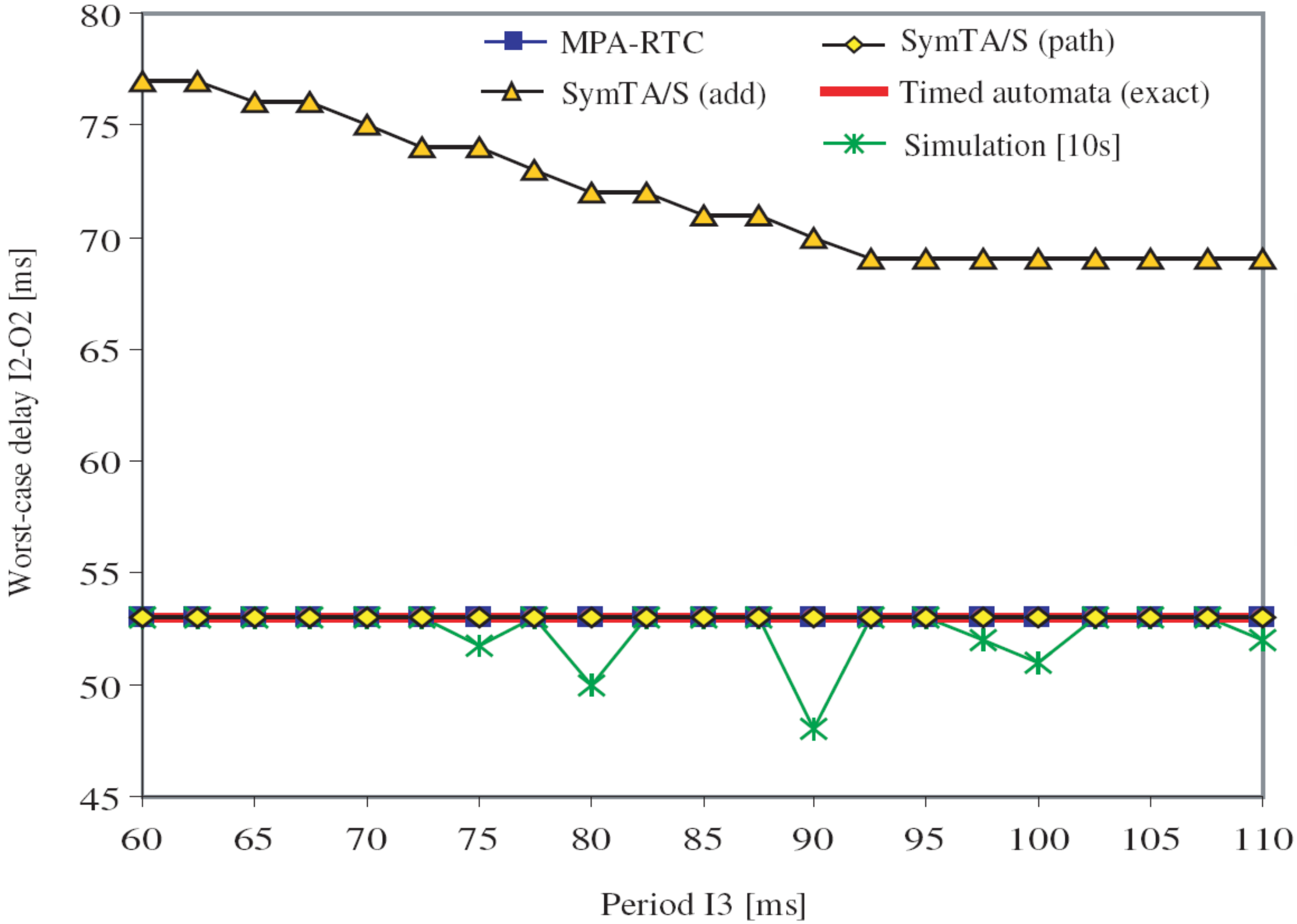


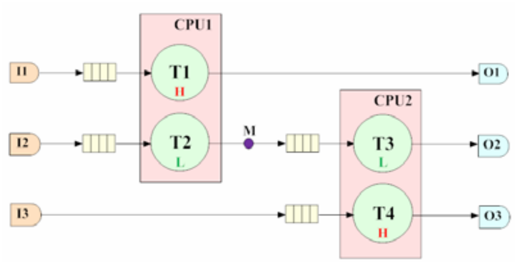
Benchmark 1 – Result interpretation

P_{I_3} = 65 ms

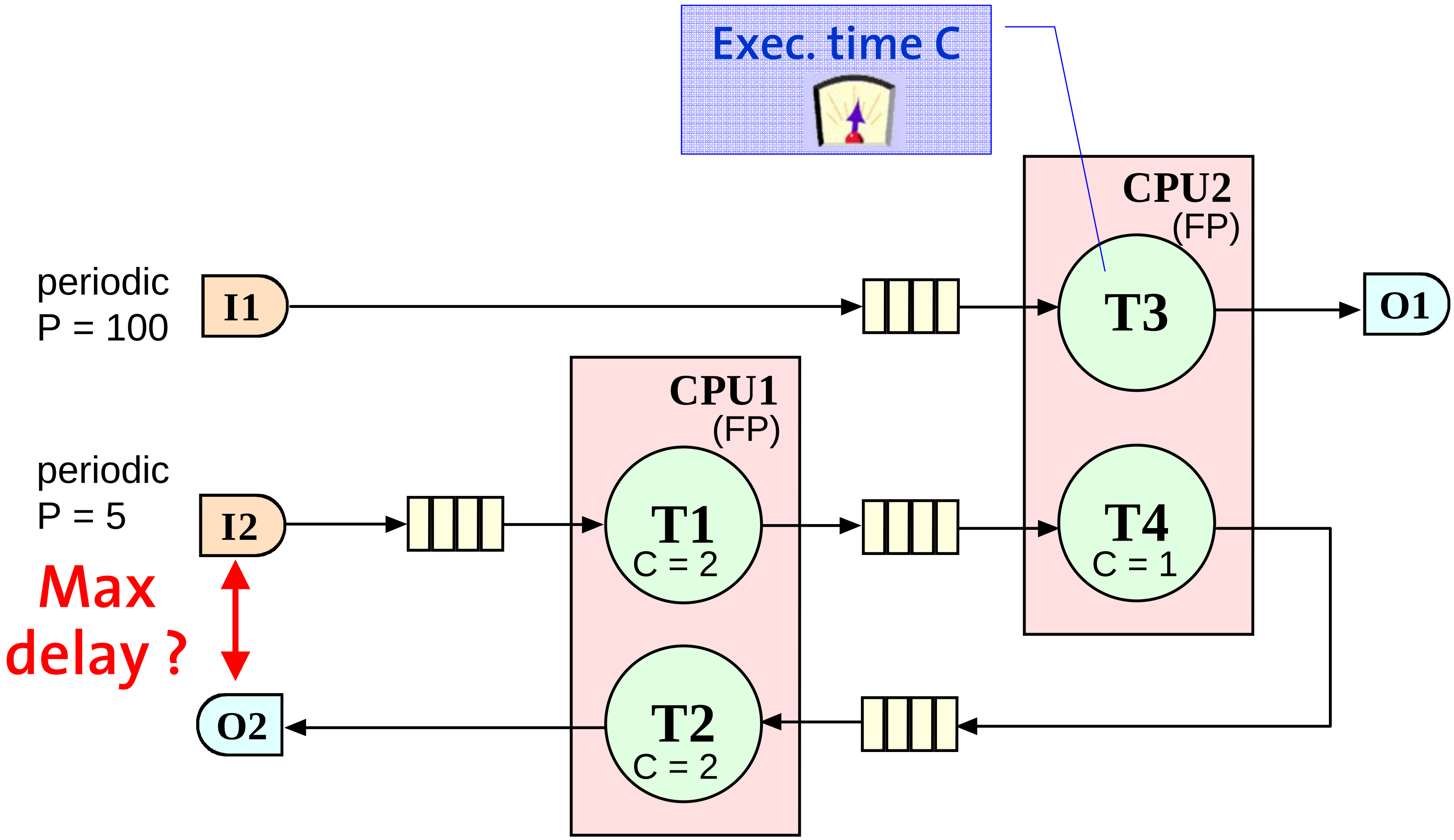


Benchmark 1 – Worst case Delay I2-O2

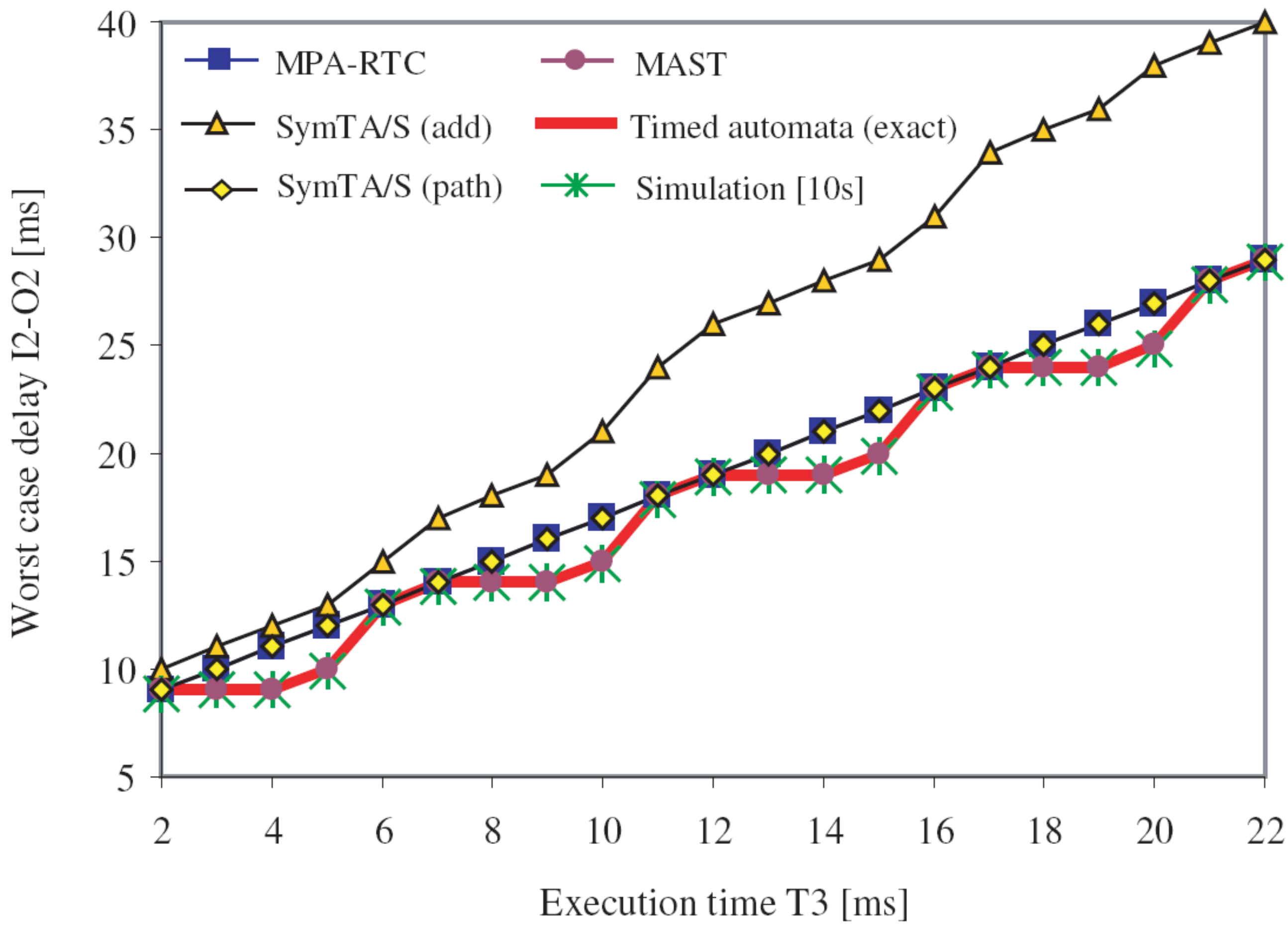


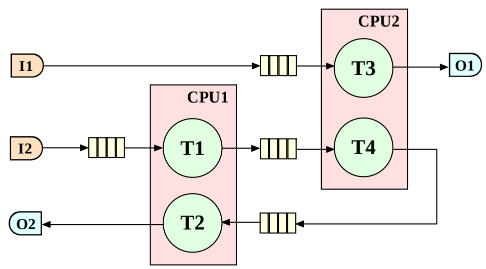


Benchmark 2 – Variable feedback

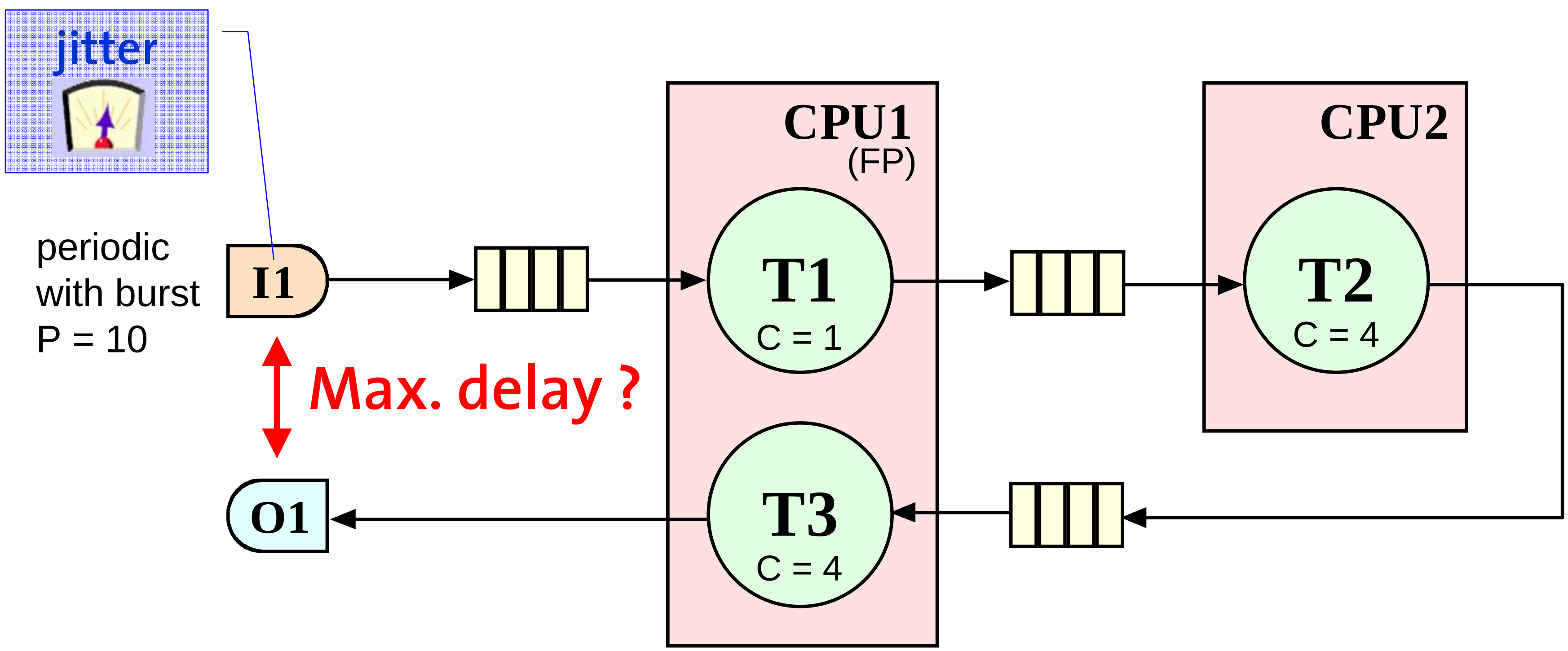


Benchmark 2 – Analysis results



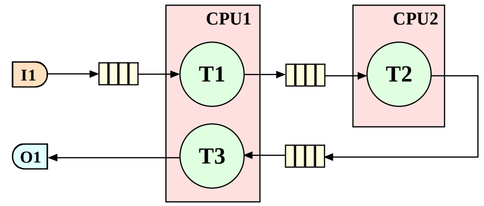


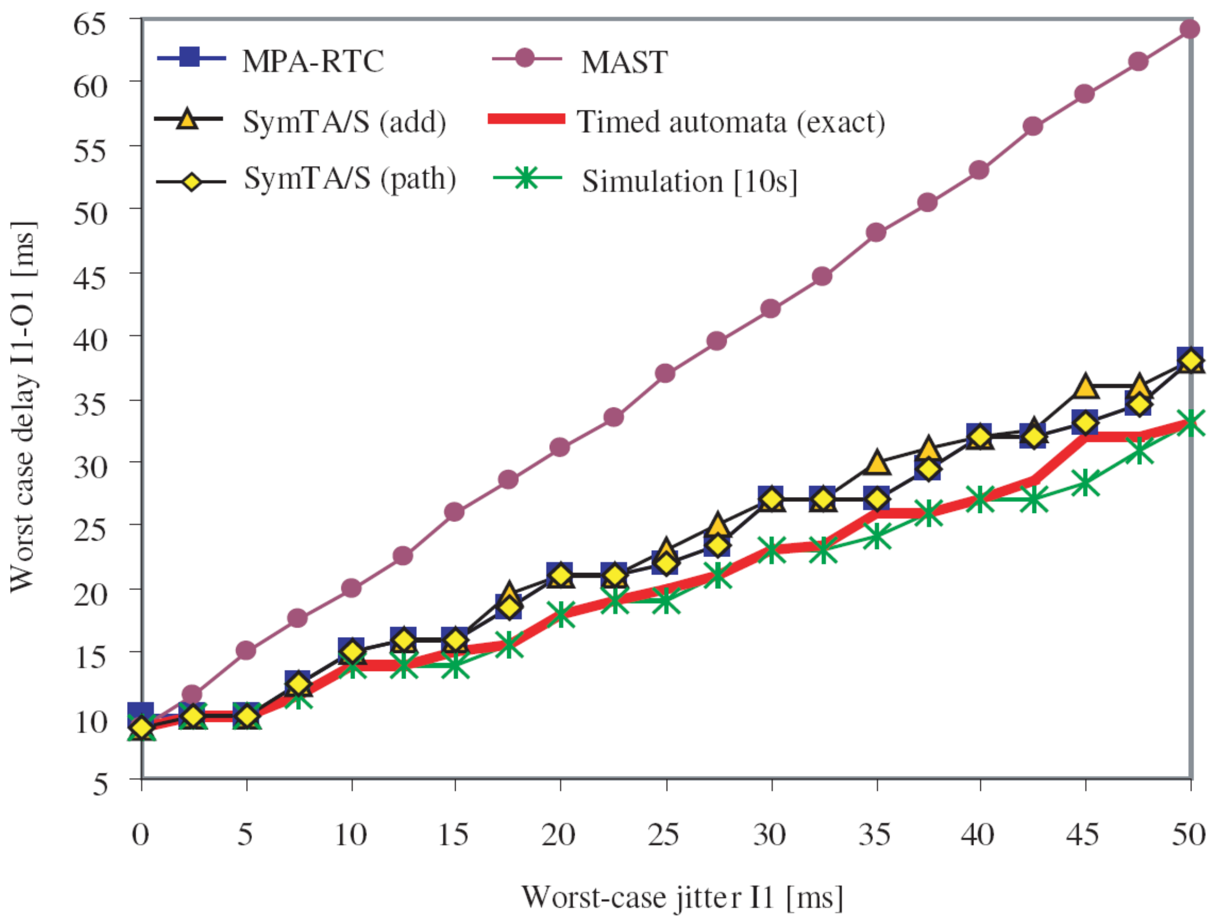
Benchmark 3 – Cyclic dependencies



Benchmark 3 – Analysis results

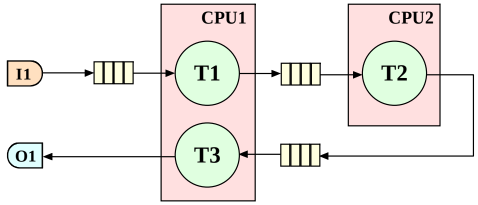
Scenario 1: priority T1 = high
priority T3 = low

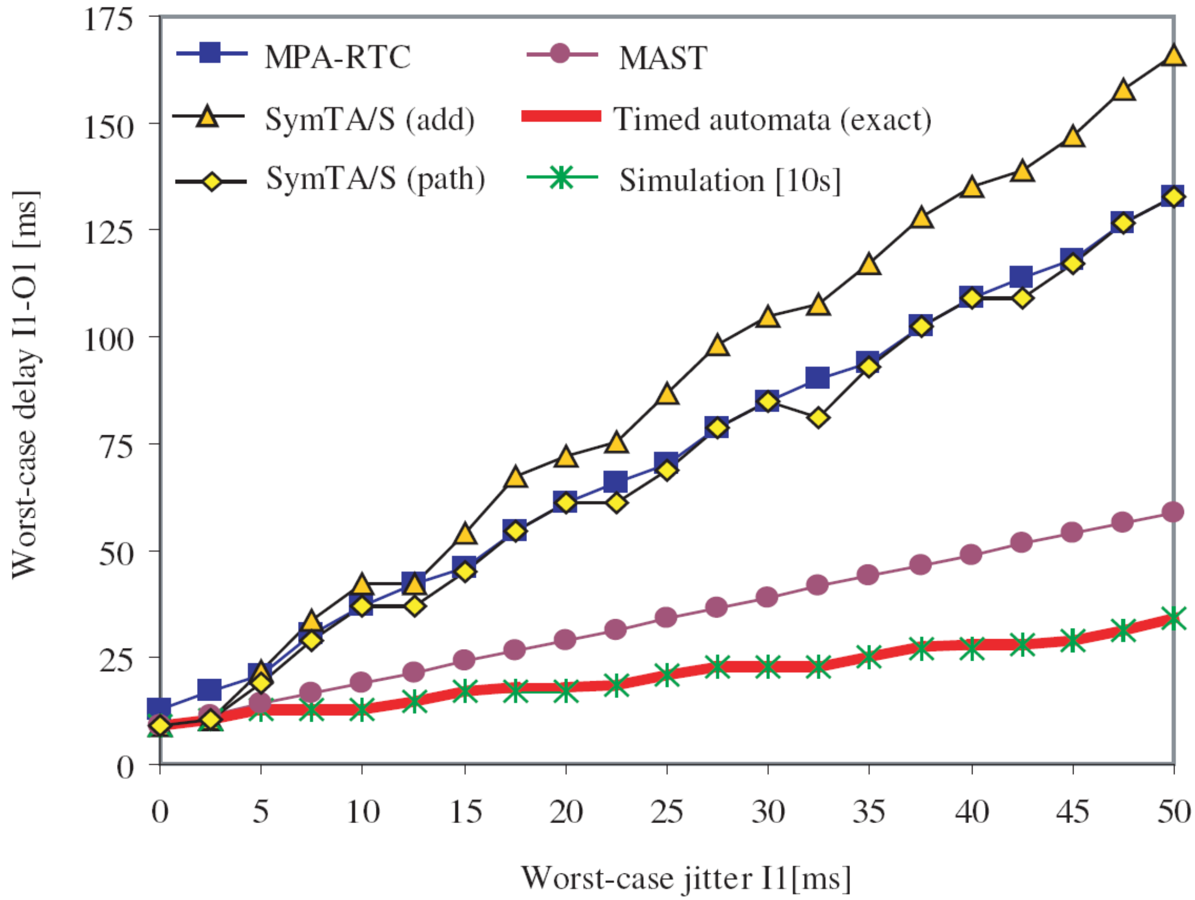




Benchmark 3 – Analysis results

Scenario 2: priority T1 = low
priority T3 = high





Analysis times [s]

		B1	B2	B3 (sc.1)	B3 (sc.2)	B4
MPA-RTC	min	0.60	0.03	0.01	0.04	0.03
	med	1.06	0.04	0.01	0.15	0.05
	max	19.72	0.08	0.04	0.30	0.20
SymTA/S	min	0.05	0.03	0.03	0.03	0.06
	med	0.09	0.05	0.06	0.34	0.09
	max	1.50	0.23	0.09	0.80	0.31
MAST	min	-	< 0.5	< 0.5	< 0.5	< 0.5
	med	-	< 0.5	< 0.5	< 0.5	< 0.5
	max	-	< 0.5	< 0.5	< 0.5	< 0.5
Timed aut.	min	18.0	< 0.5	< 0.5	< 0.5	< 0.5
	med	34.5	< 0.5	1.0	< 0.5	< 0.5
	max	60.5	< 0.5	52.0	5.5	< 0.5
Simulation	min	1.0	< 0.5	0.5	0.5	< 0.5
	med	1.0	< 0.5	0.5	0.5	< 0.5
	max	1.0	< 0.5	0.5	0.5	< 0.5

Outline

- Motivation
- Abstractions
- Benchmarks
- Conclusions

Discussion

- Approximation of complex event streams with standard event models can lead to poor performance predictions at local level
- Holistic approaches better in the presence of correlations among task activations (e.g. data dependencies)
- Cyclic dependencies represent a serious pitfall for the accuracy of compositional analysis methods
- Holistic methods less appropriate for timing properties referred to the *actual* release time of an event within a large jitter interval

Conclusions

- The analysis accuracy and the analysis time depend highly on the specific system characteristics
- None of the analysis methods performed best in all benchmarks
- The analysis results of the different approaches are remarkable different even for apparently basic systems
- The choice of an appropriate analysis abstraction matters
- The problem to provide accurate performance predictions for general systems is still far from solved

Thank you!

Simon Perathoner

perathoner@tik.ee.ethz.ch