Modular Performance Analysis of MPSoC

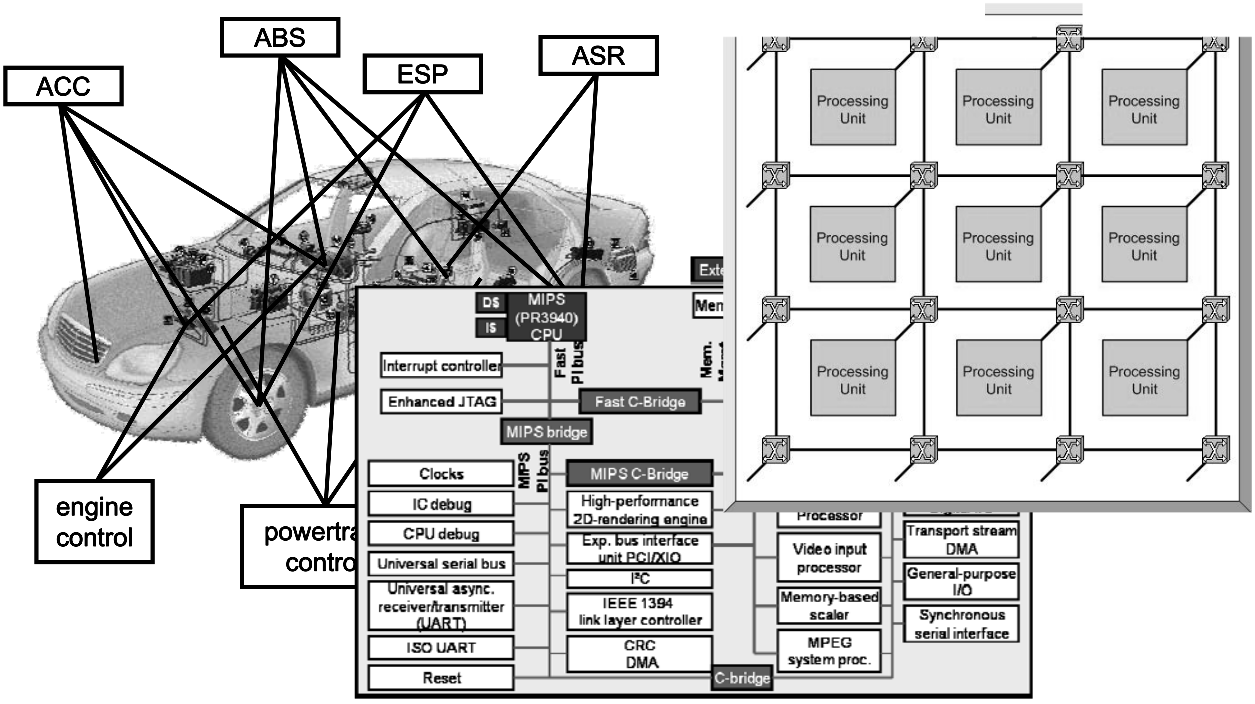
Lothar Thiele

ETH Zurich, Switzerland

Outline

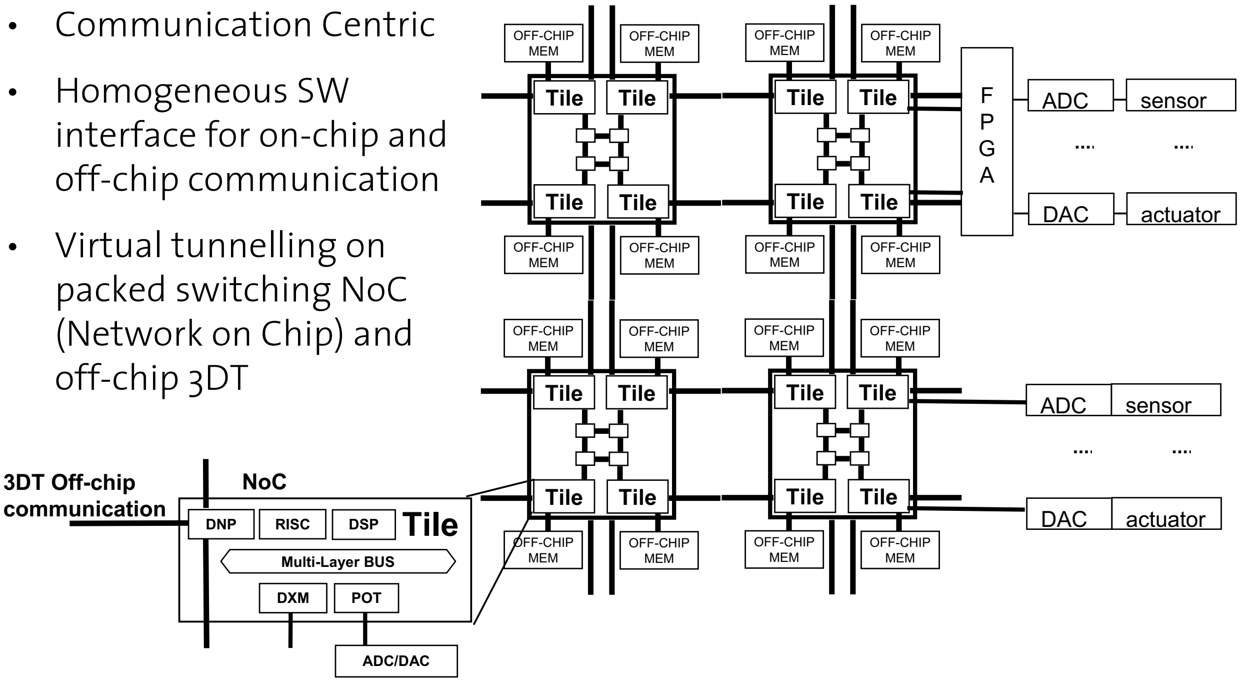
- *Embedding of Performance Analysis*
- Modular Performance Analysis
- Examples

Target Platforms



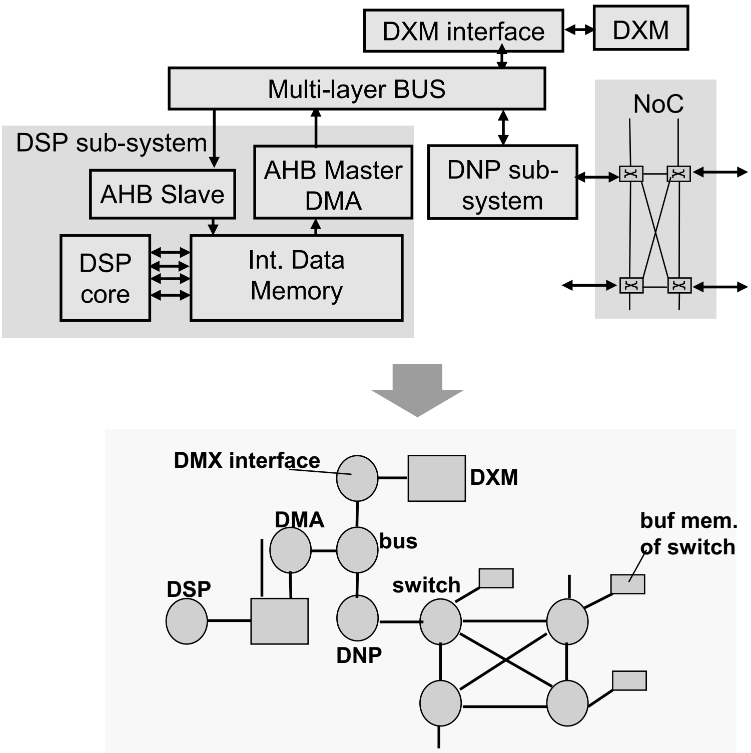
A sample HW Architecture

- Communication Centric
- Homogeneous SW interface for on-chip and off-chip communication
- Virtual tunnelling on packed switching NoC (Network on Chip) and off-chip 3DT

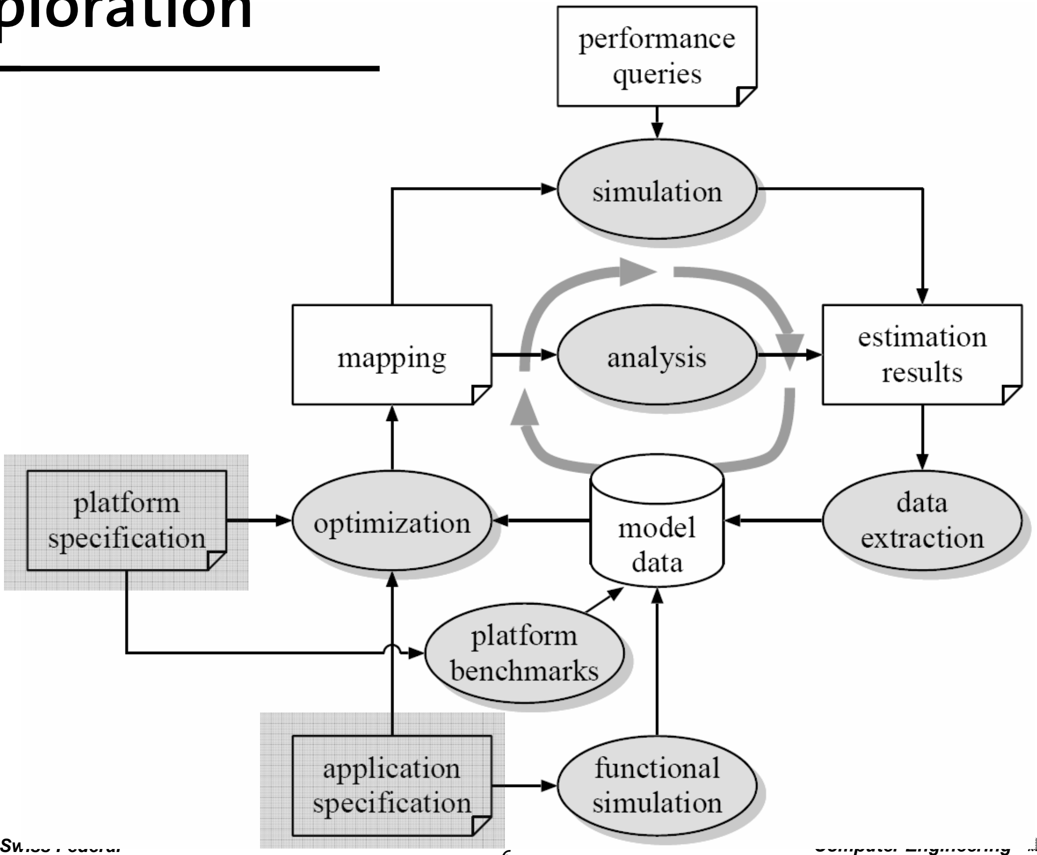


Target Platform Abstraction (1)

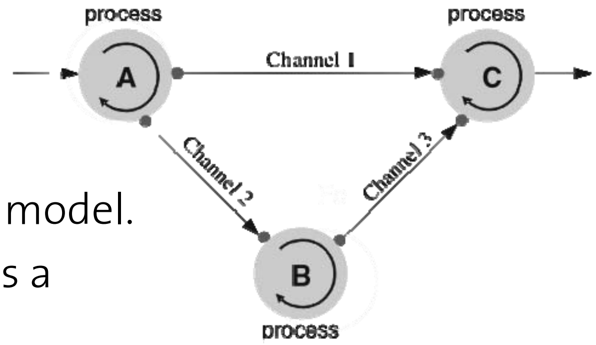
- *Topology modeled by a graph*
  - two node types:
    - execution and comm. resources
    - storage resources
- *Execution resources*
  - RISCs, DSPs, ...
- *Communication resources*
  - buses, switches, links, I/Os
- *Storage resources*
  - RAMs, HW FIFOs, ...



Exploration



Application Model

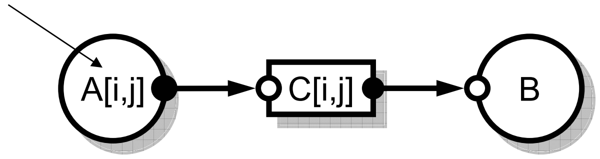


- ***Model-based design*:**
  - Stream-oriented application model.
  - The application is modeled as a network of processes
  - Processes communicate via unidirectional channels.
  - Kahn PN: Determinate (functional properties are independent of scheduling).

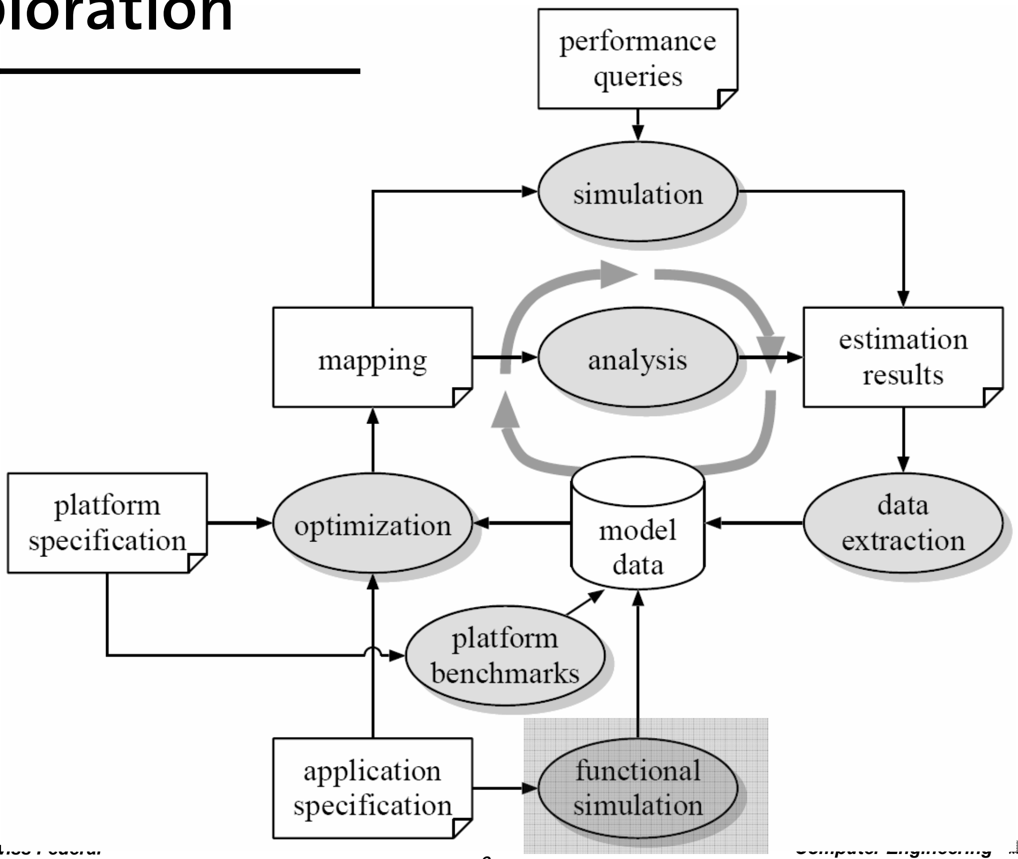
Scalability at Specification Level

- ***Separation*** of instruction level parallelism (inside processes) and task-level parallelism.
- Use of ***iterators*** in
  - architecture specification
  - application specification
  - mapping specification

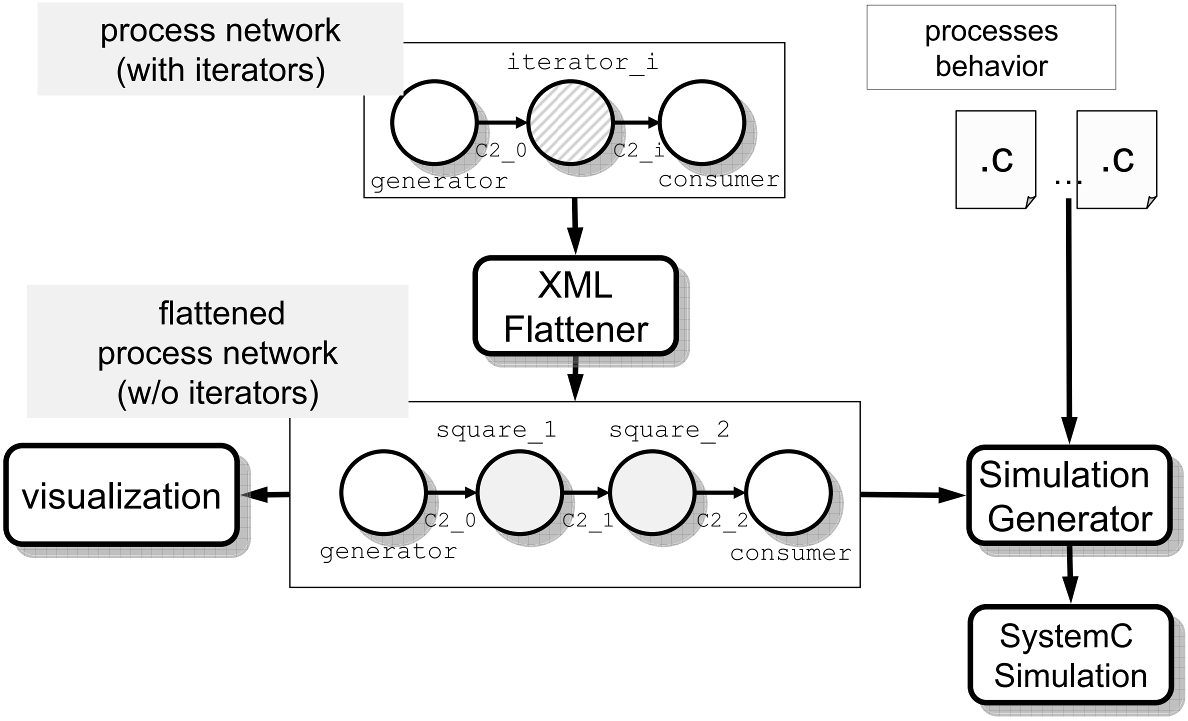
$$\{(i,j) : 1 \leq i \leq N \wedge i \leq j \leq N\}$$



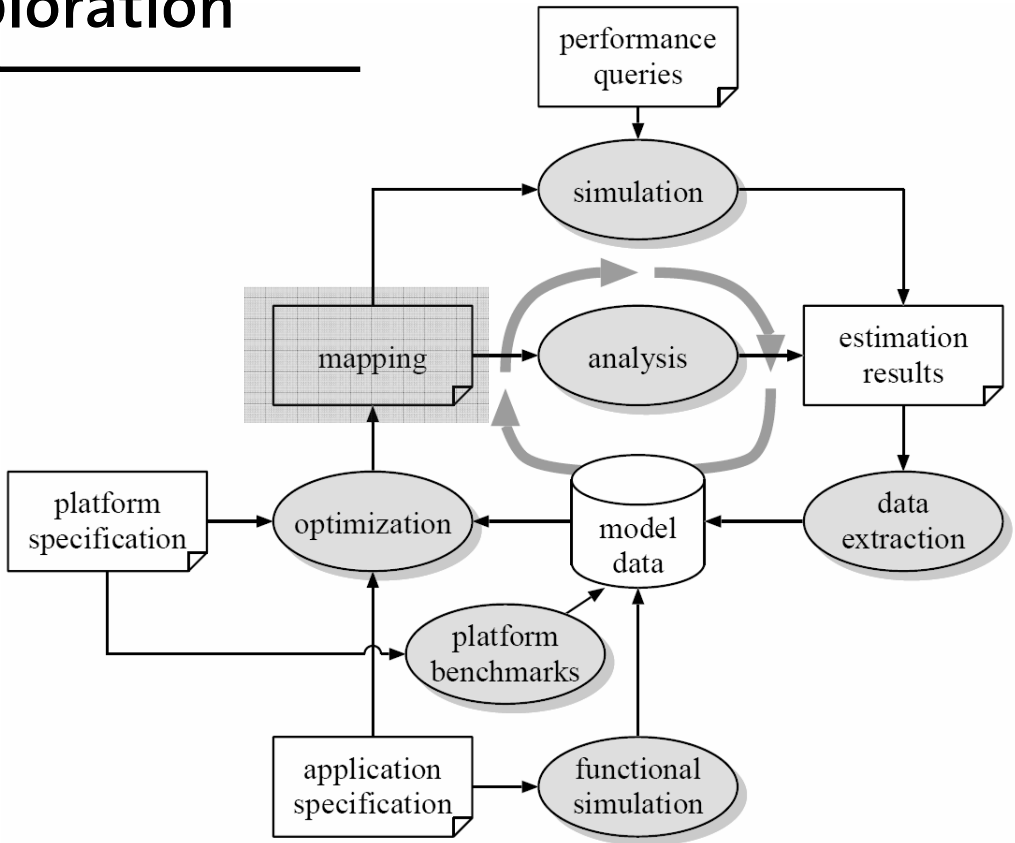
Exploration



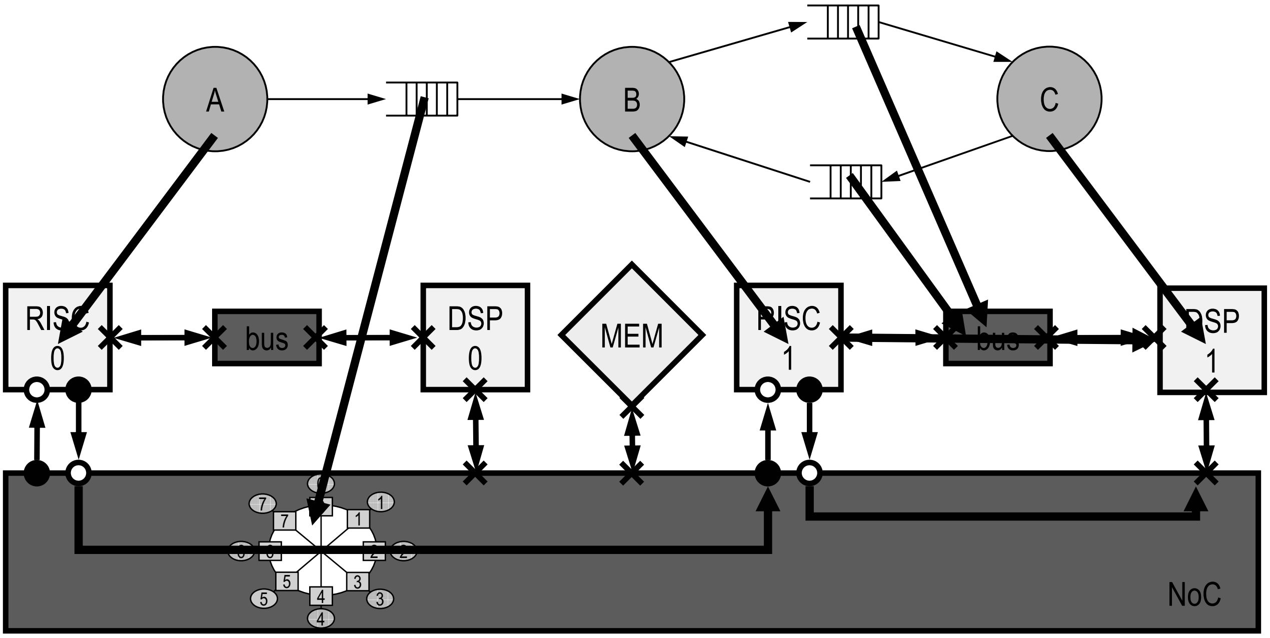
Application Functional Simulation



Exploration

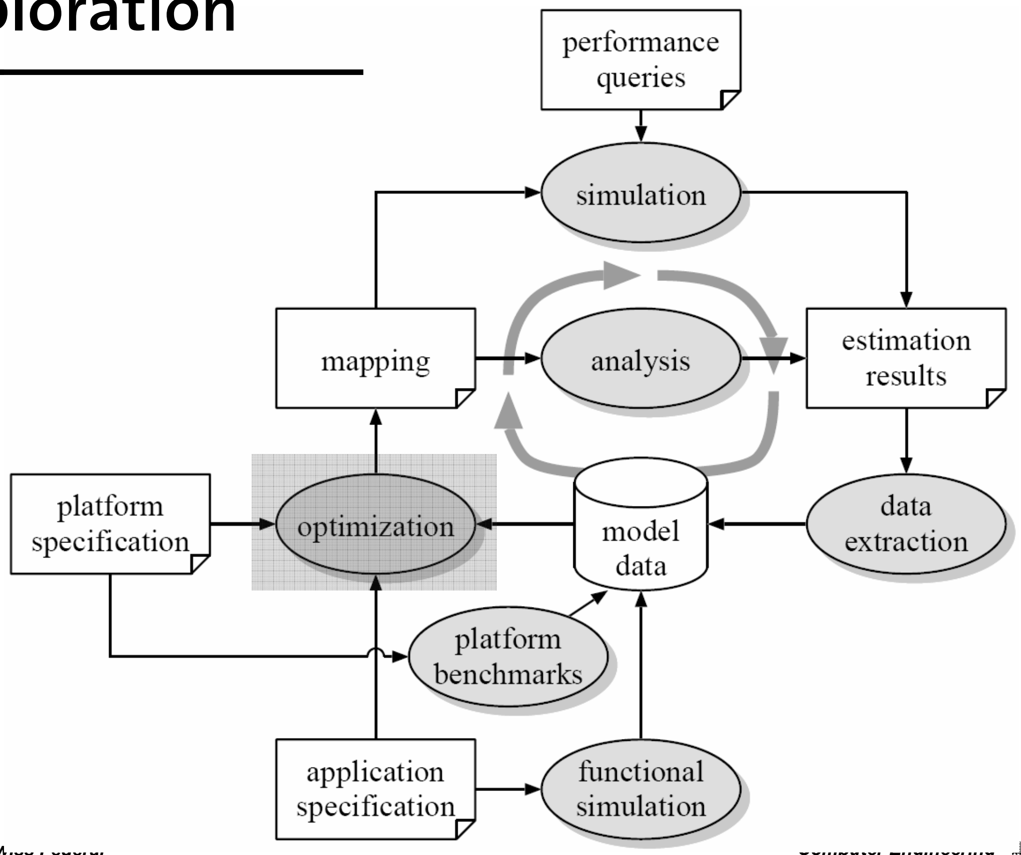


Mapping

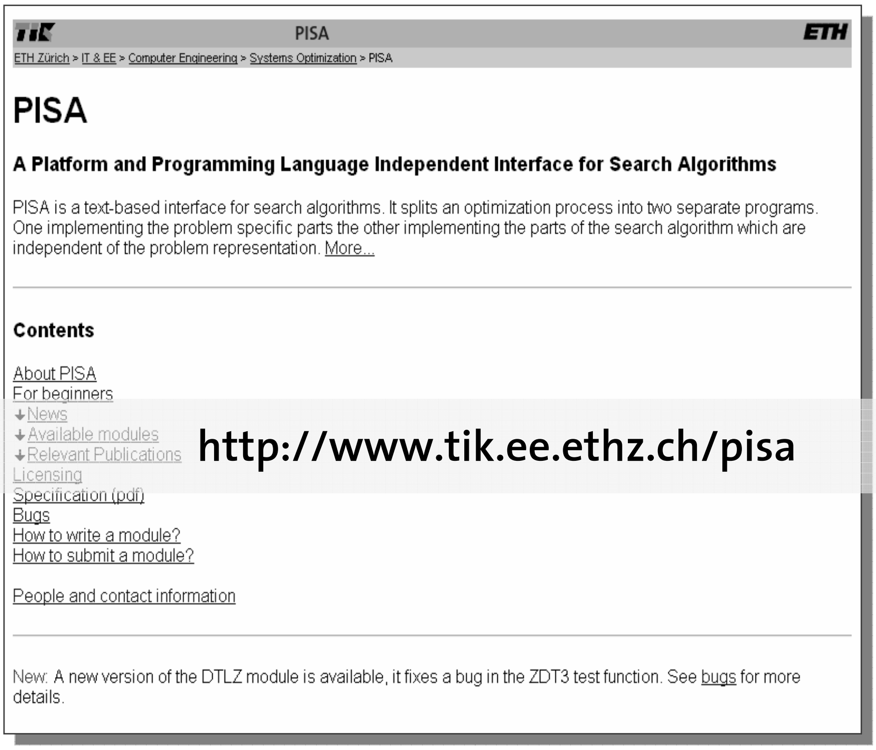


Mapping = binding + scheduling

Exploration



Mapping Optimization



ETH Zürich > IT & EE > Computer Engineering > Systems Optimization > PISA

PISA

A Platform and Programming Language Independent Interface for Search Algorithms

PISA is a text-based interface for search algorithms. It splits an optimization process into two separate programs. One implementing the problem specific parts the other implementing the parts of the search algorithm which are independent of the problem representation. More...

Contents

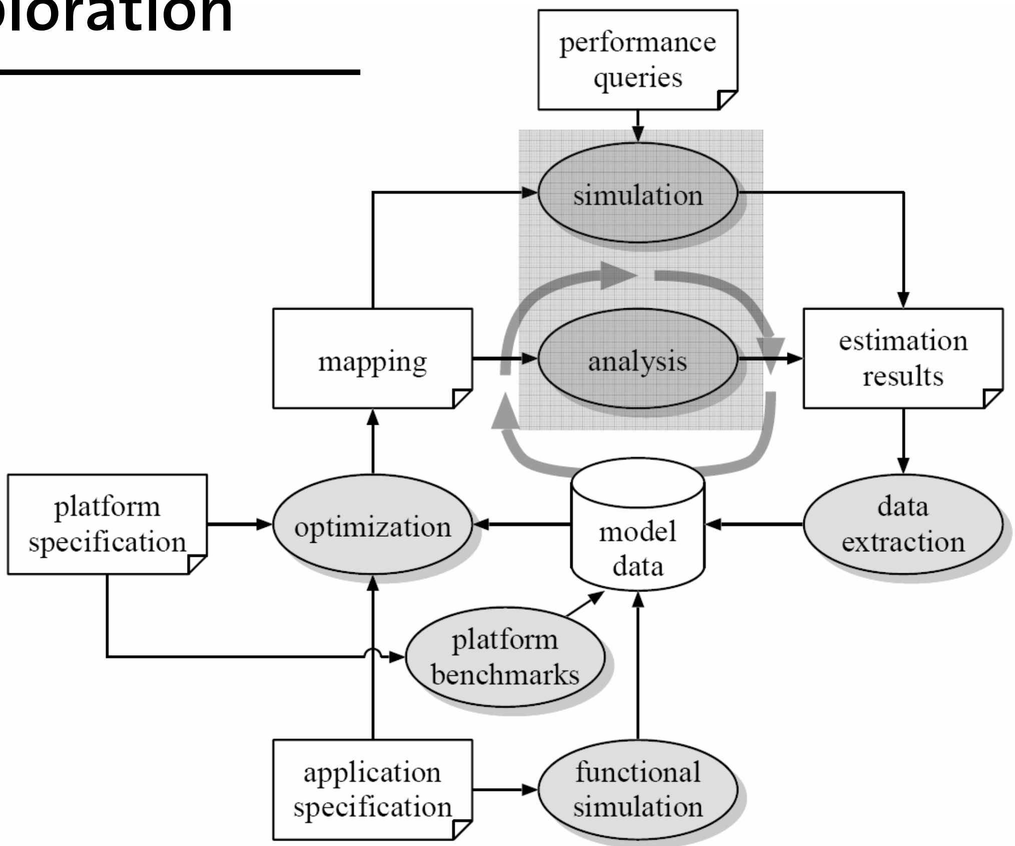
About PISA
For beginners
↓News
↓Available modules
↓Relevant Publications
Licensing
Specification (pdf)
Bugs
How to write a module?
How to submit a module?

People and contact information

New: A new version of the DTLZ module is available, it fixes a bug in the ZDT3 test function. See bugs for more details.

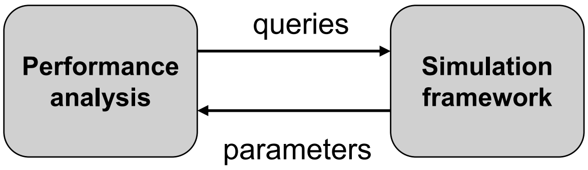
http://www.tik.ee.ethz.ch/pisa

Exploration



Performance Estimation

- ***Layers of abstraction***:
  - Simulation
    - Use for complete system validation
    - Use for getting parameters of single components
  - Trace-based performance analysis
  - Analytic methods
    - Back-of-the-envelope
    - Modular performance analysis MPA: www.mpa.ethz.ch



Back-of-the-envelope Analysis

processor c with worst total runtime

number of firings of task p

runtime of task p on processor c

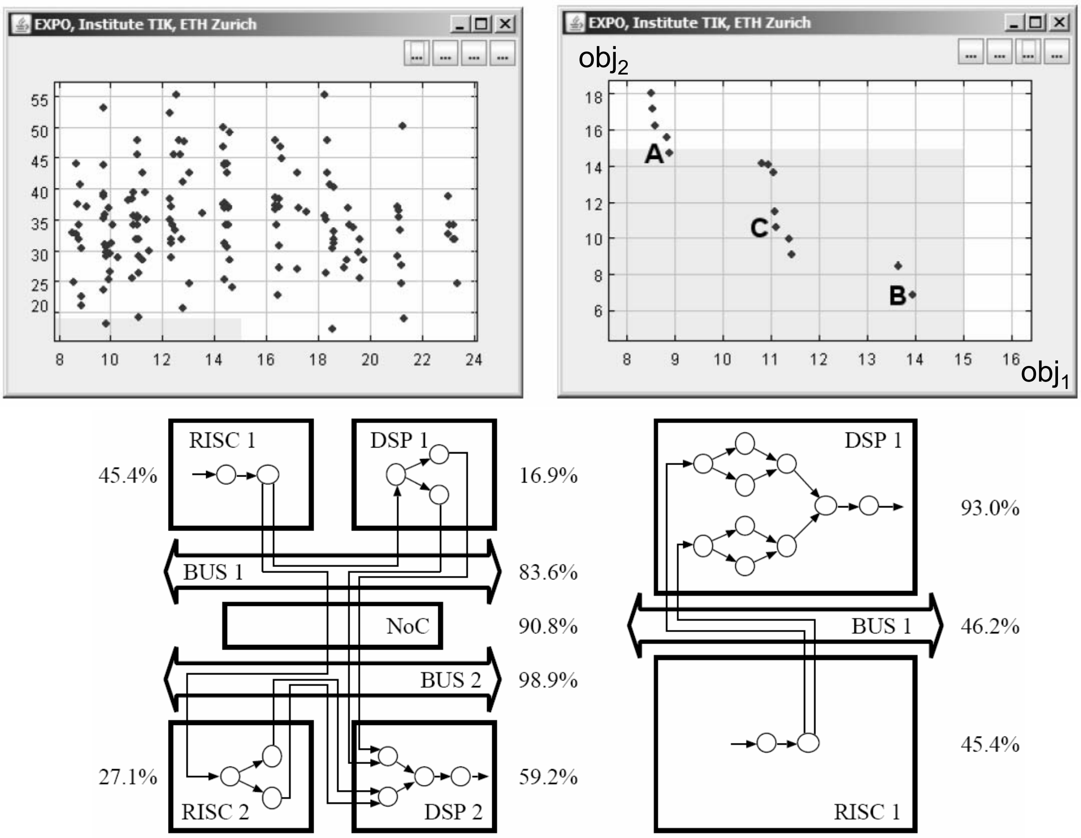
$$obj_1 = \max_{c \in \mathcal{C}} \left\{ \sum_{\forall p \text{ mapped to } c} n(p) \cdot r(p,c) \right\}$$

communication link with worst load

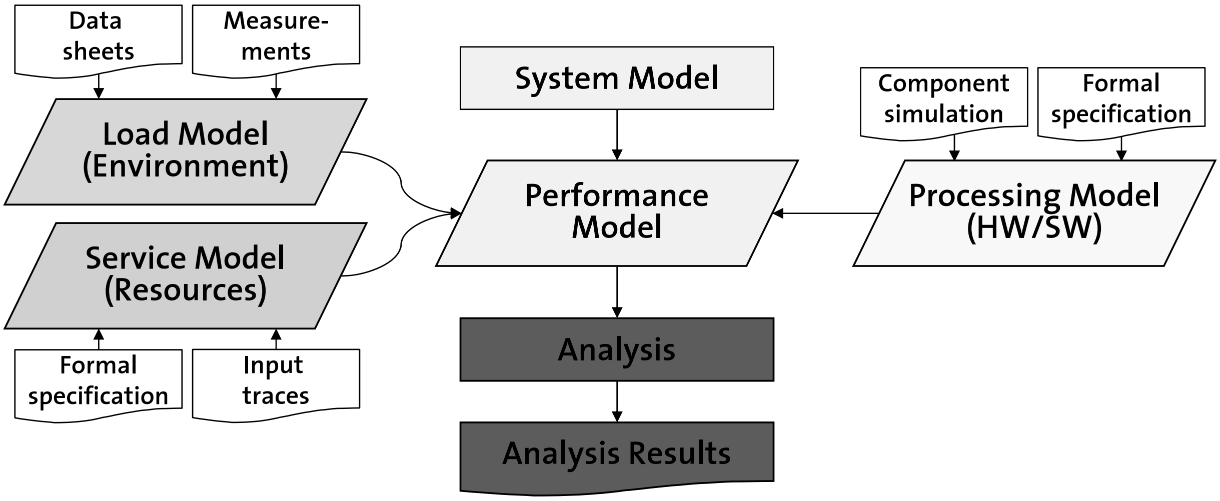
communication request from channel s

bandwidth of communication link g

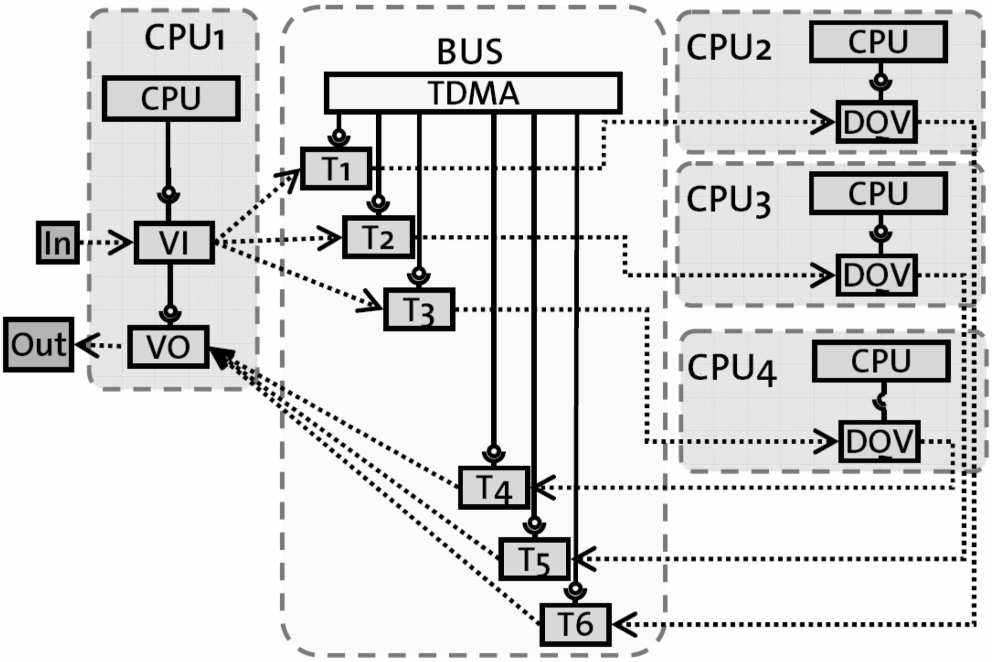
$$obj_2 = \max_{g \in \mathcal{G}} \left\{ \sum_{\forall s \text{ mapped onto } g} \frac{b(s)}{t(g)} \right\}$$



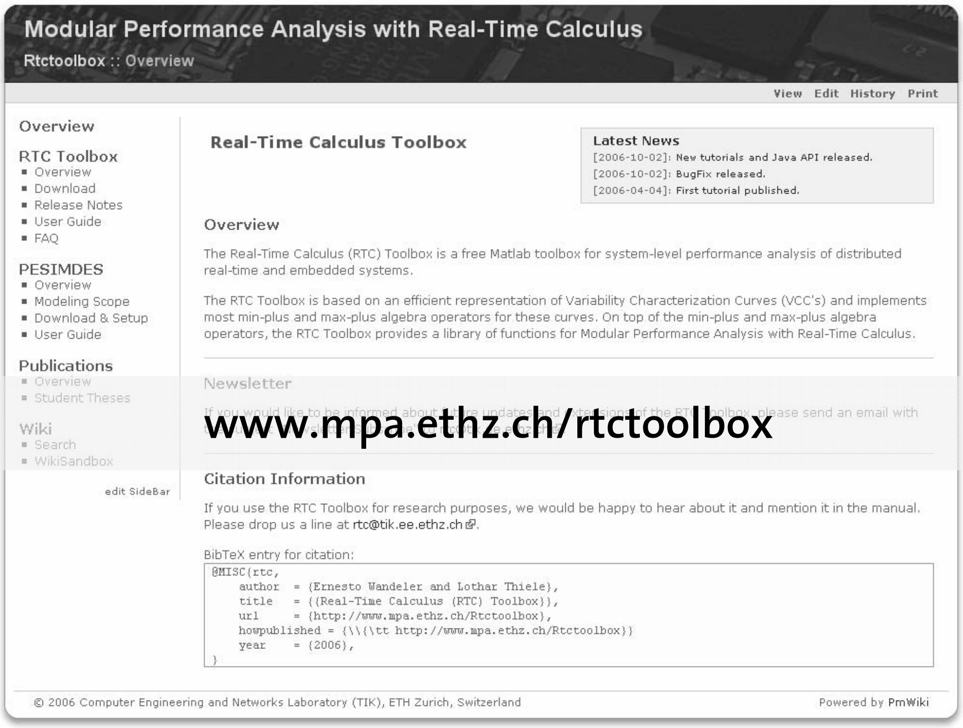
Modular Performance Analysis (MPA)



MPA Performance Model



MPA (Modular Performance Analysis)



Modular Performance Analysis with Real-Time Calculus

Rtctoolbox :: Overview

Overview

RTC Toolbox
- Overview
- Download
- Release Notes
- User Guide
- FAQ

PESIMDES
- Overview
- Modeling Scope
- Download & Setup
- User Guide

Publications
- Overview
- Student Theses

Wiki
- Search
- WikiSandbox

edit SideBar

Real-Time Calculus Toolbox

Latest News
[2006-10-02]: New tutorials and Java API released.
[2006-10-02]: BugFix released.
[2006-04-04]: First tutorial published.

Overview

The Real-Time Calculus (RTC) Toolbox is a free Matlab toolbox for system-level performance analysis of distributed real-time and embedded systems.

The RTC Toolbox is based on an efficient representation of Variability Characterization Curves (VCC's) and implements most min-plus and max-plus algebra operators for these curves. On top of the min-plus and max-plus algebra operators, the RTC Toolbox provides a library of functions for Modular Performance Analysis with Real-Time Calculus.

Newsletter

If you would like to be informed about future updates and extensions of the RTC Toolbox, please send an email with the subject "Newsletter Subscribe" to rtc@tik.ee.ethz.ch.

Citation Information

If you use the RTC Toolbox for research purposes, we would be happy to hear about it and mention it in the manual. Please drop us a line at rtc@tik.ee.ethz.ch.

BibTeX entry for citation:

```
@MISC{rtc,
    author  = {Ernesto Wandeler and Lothar Thiele},
    title   = {{Real-Time Calculus (RTC) Toolbox}},
    url     = {http://www.mpa.ethz.ch/Rtctoolbox},
    howpublished = {\\{\tt http://www.mpa.ethz.ch/Rtctoolbox}}
    year    = {2006},
}
```

© 2006 Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Switzerland

Powered by PmWiki

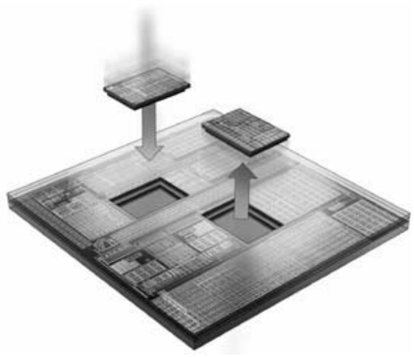
www.mpa.ethz.ch/rtctoolbox

Analysis and Design

Embedded System =

Computation + Resource Interaction

Analysis:
   Infer system properties from subsystem properties.

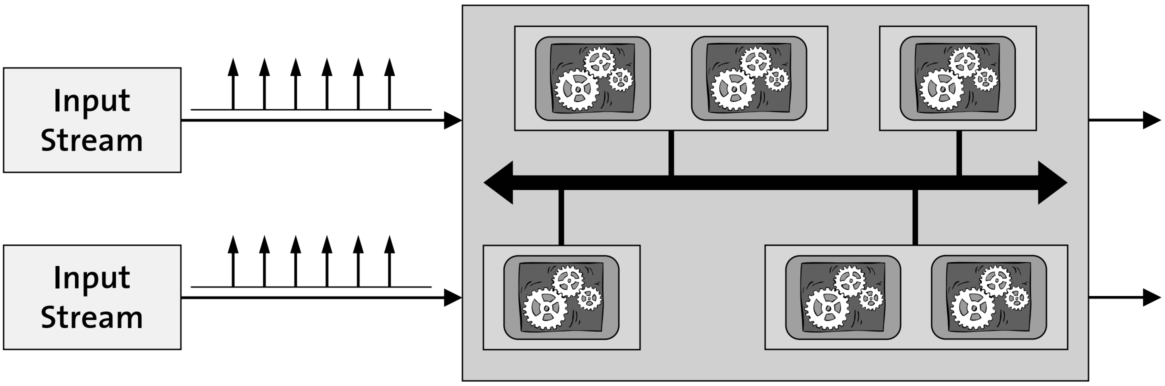


Design:
   Build a system from subsystems while meeting requirements.

Outline

- Embedding of Performance Analysis
- *Modular Performance Analysis*
- Examples

System-Level Performance Analysis



Memory Requirements?

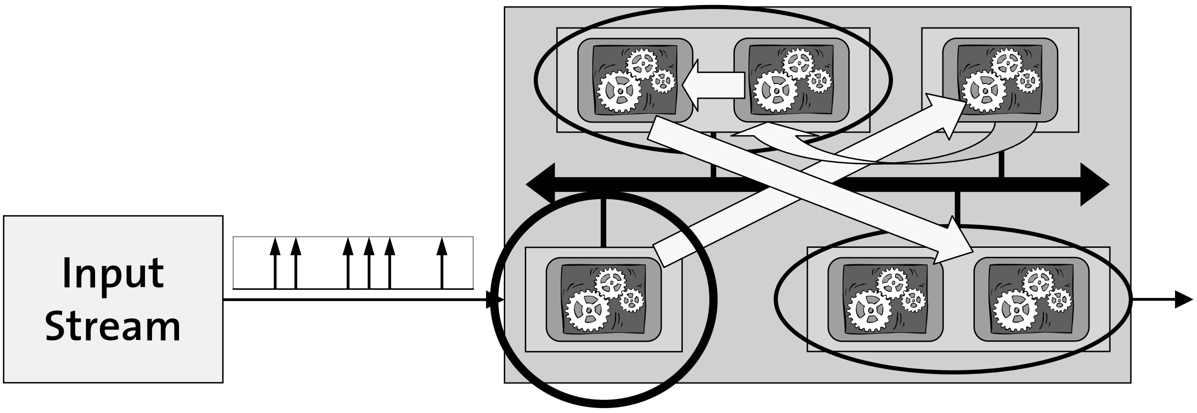
Timing Properties?

Bottleneck?

Processor Speeds?

Bus Utilization?

Difficulties

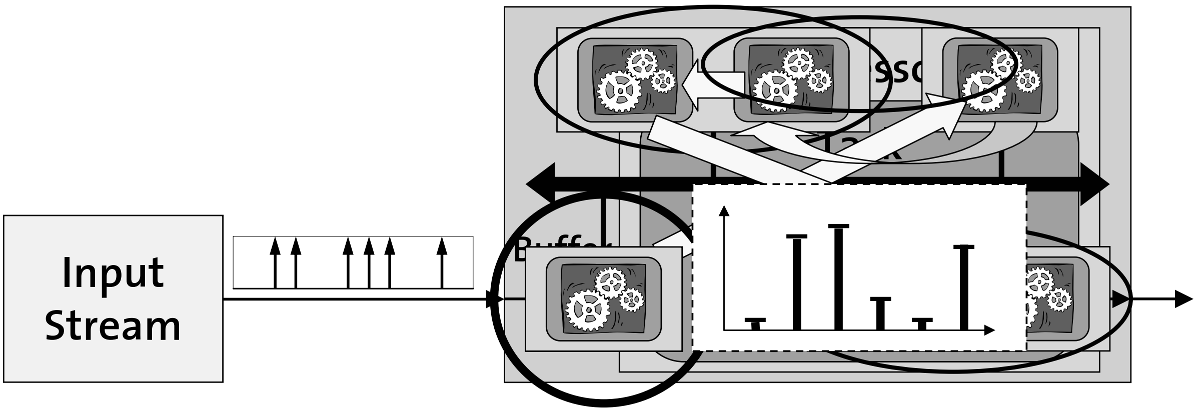


Interference Communication

Interference Computation

Nondeterministic Environment

Difficulties



Interference Communication

Interference Computation

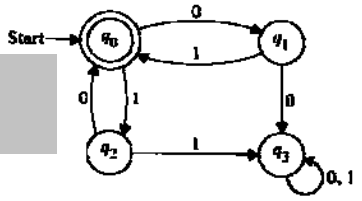
Nondeterministic Environment

Complex Resource Availability

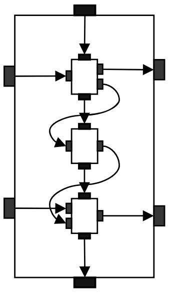
Complex Execution Demand

What is necessary?

From interfaces that talk about static types

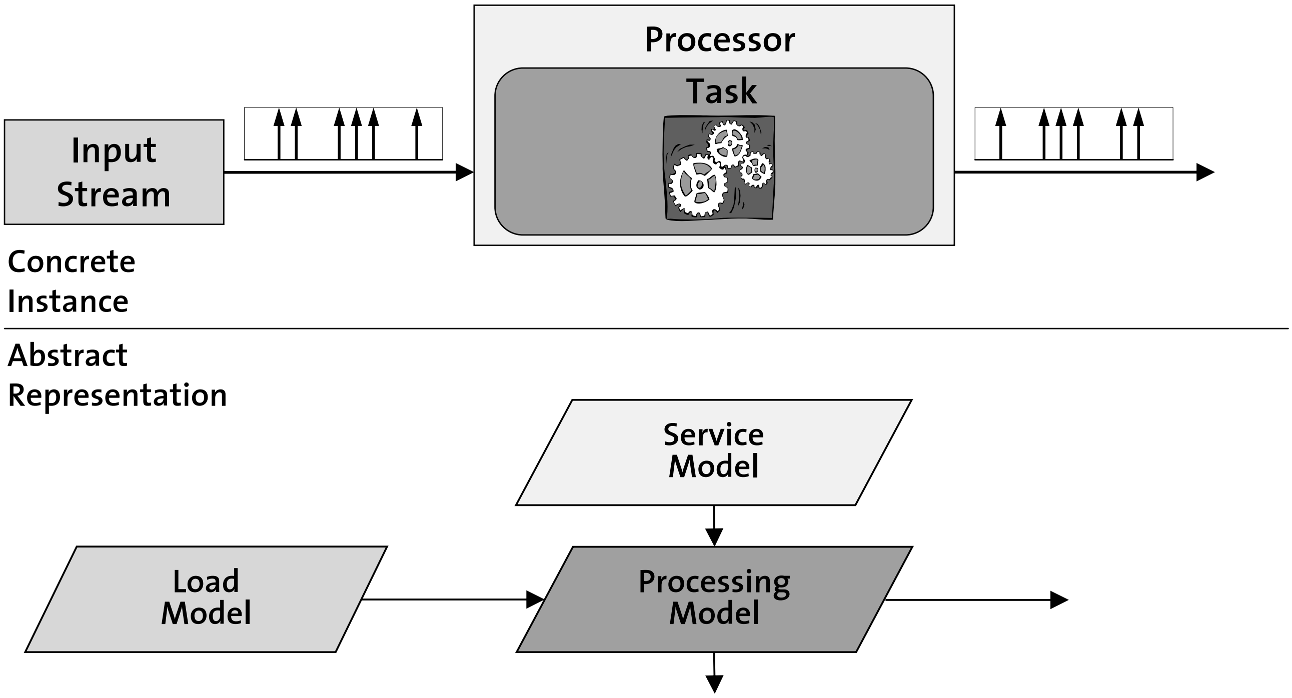


via behavioral types



towards resource types

Abstract Models for Performance Analysis



Processor

Task

Input Stream

Concrete Instance

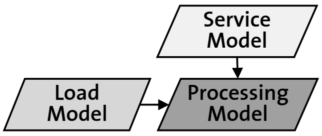
Abstract Representation

Service Model

Load Model

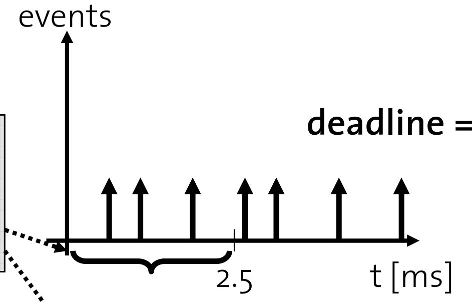
Processing Model

Load Model (Environment)



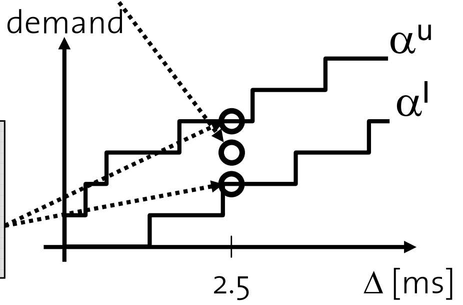
Event Stream

number of events in in t=[0 .. 2.5] ms

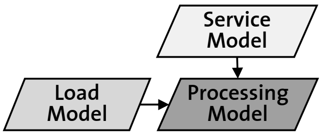


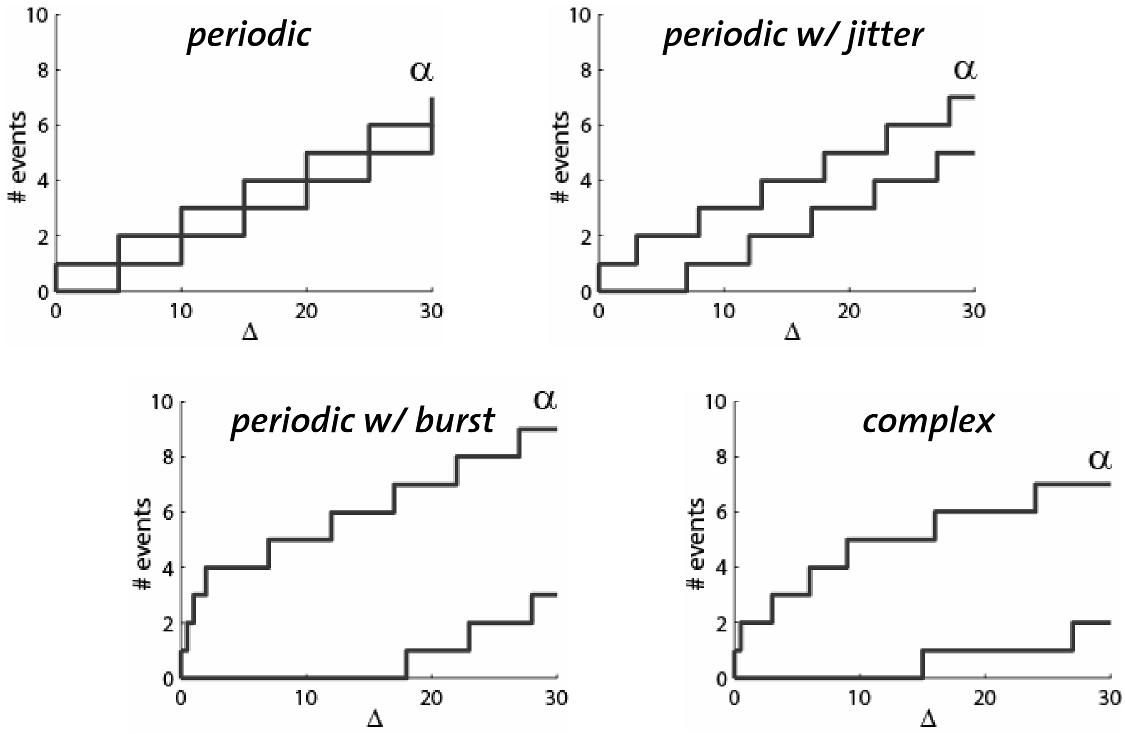
Arrival Curve α

maximum / minimum arriving demand in *any interval* of length 2.5 ms

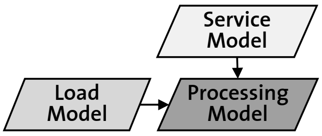


Load Model - Examples

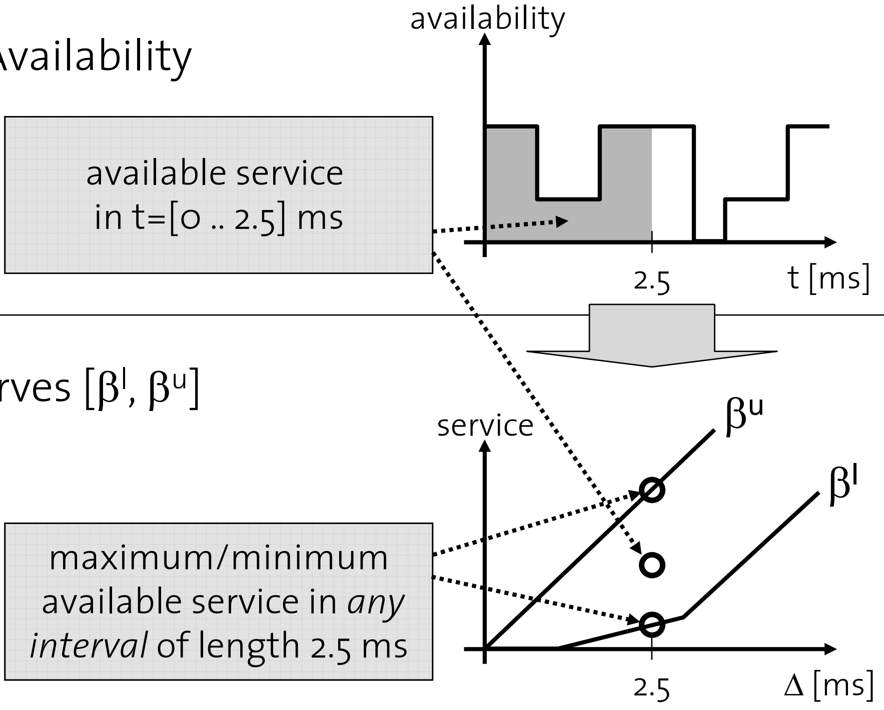




Service Model (Resources)



Resource Availability

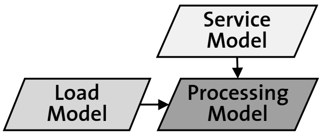


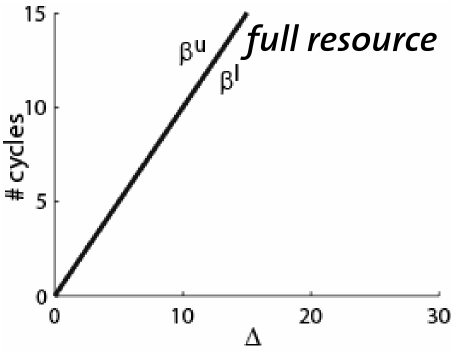
available service in t=[0 .. 2.5] ms

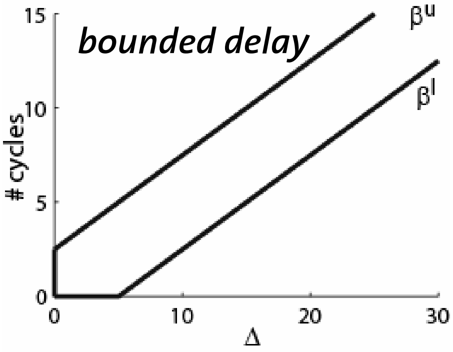
Service Curves [β^l, β^u]

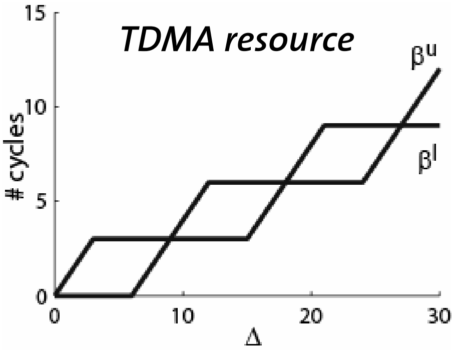
maximum/minimum available service in *any interval* of length 2.5 ms

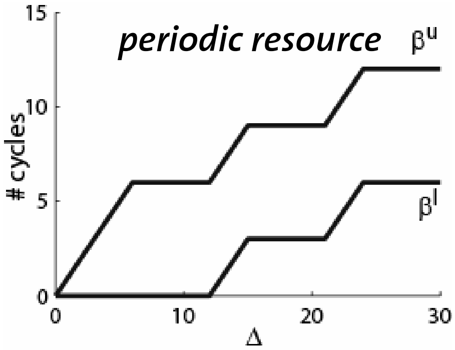
Service Model - Examples



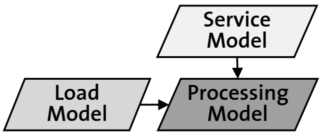






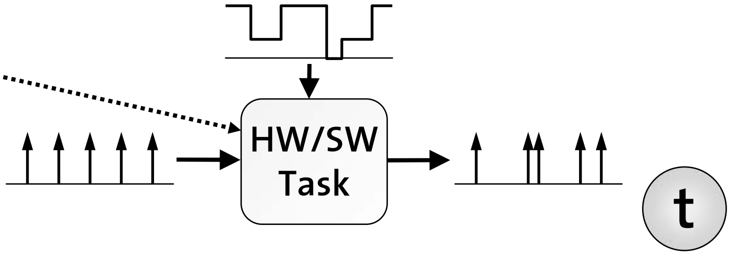


Processing Model (HW/SW)



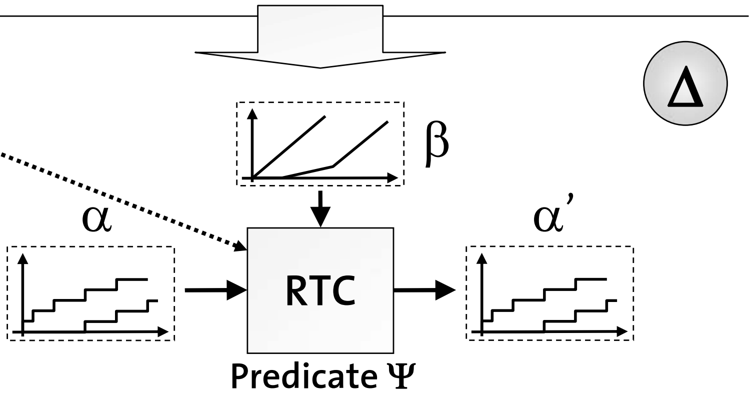
HW/SW Components

Processing semantics and functionality of HW/SW tasks

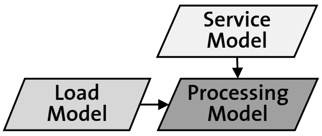


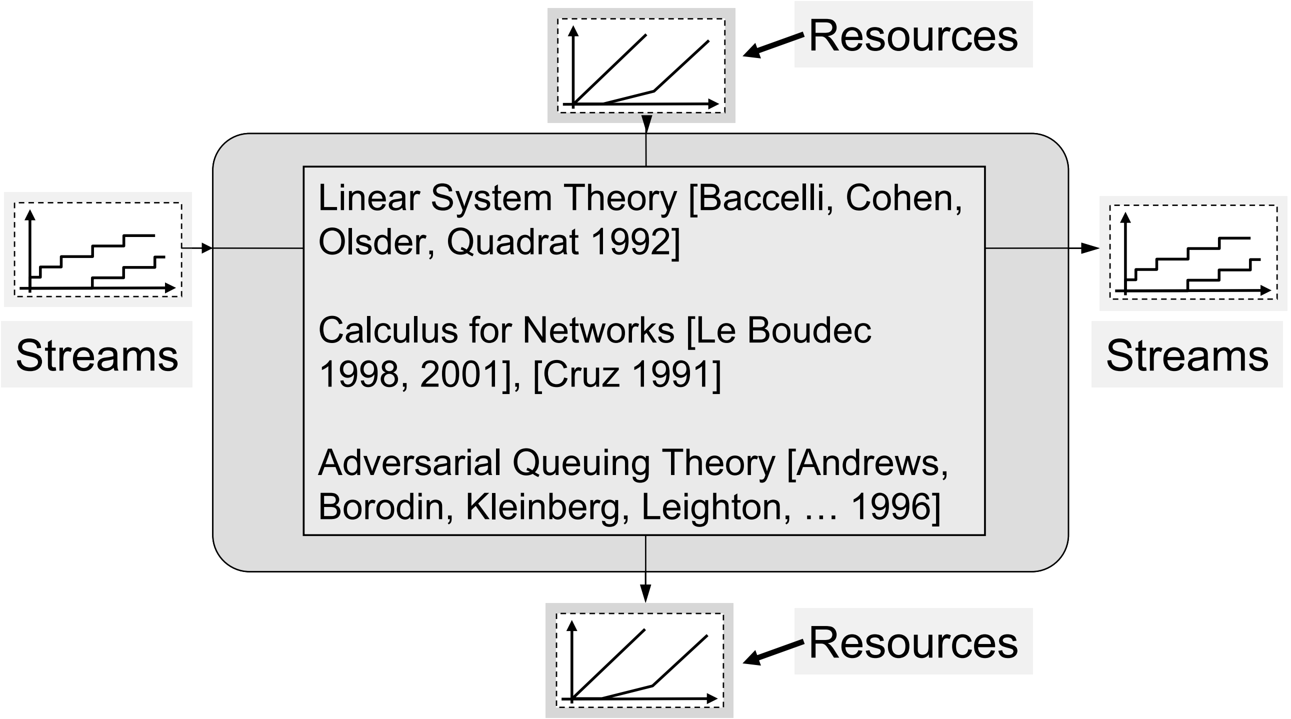
Abstract Components

$$\alpha'(\Delta) = f(\alpha, \beta)$$



System Module





Linear System Theory [Baccelli, Cohen, Olsder, Quadrat 1992]

Calculus for Networks [Le Boudec 1998, 2001], [Cruz 1991]

Adversarial Queuing Theory [Andrews, Borodin, Kleinberg, Leighton, … 1996]

Min-Plus Algebra

$(\mathbb{R} \cup \{+\infty\}, \wedge, +)$: min-plus algebra

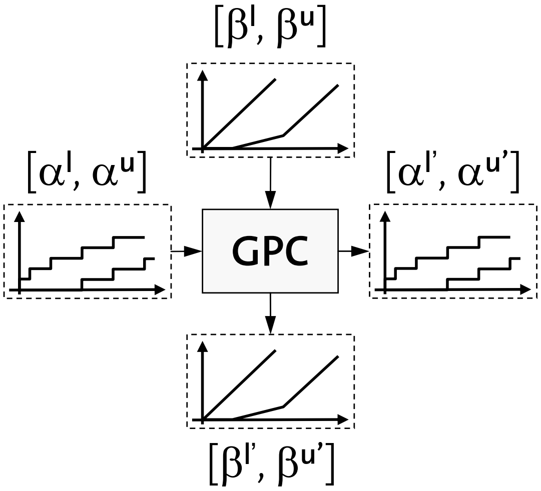
Min-plus convolution and de-convolution:

$$(f \underline{\otimes} g)(t) = \inf_{0 \leq u \leq t} \{f(t-u) + g(u)\}$$

$$(f \overline{\oslash} g)(t) = \sup_{u \geq 0} \{f(t+u) - g(u)\}$$

Processing Model – Examples

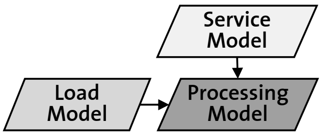
Greedy Processing Component



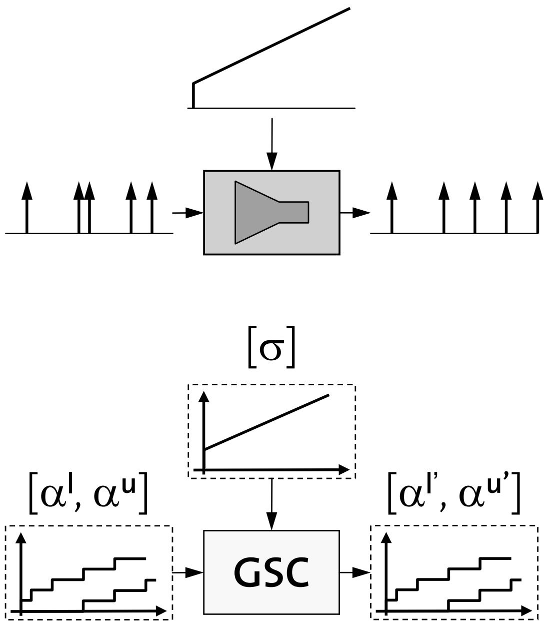
Real-Time Calculus

$$\begin{aligned} \alpha'^{u} &= \min\{(\alpha^{u} \otimes \beta^{u}) \oslash \beta^{l}, \beta^{u}\} \\ \alpha'^{l} &= \min\{(\alpha^{l} \oslash \beta^{u}) \otimes \beta^{l}, \beta^{l}\} \\ \beta'^{u} &= (\beta^{u} - \alpha^{l}) \overline{\oslash}\, 0 \\ \beta'^{l} &= (\beta^{l} - \alpha^{u}) \overline{\otimes}\, 0 \end{aligned}$$

Processing Model – Examples



Greedy Shaper Component



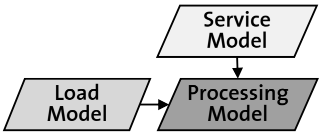
Behavioral Description

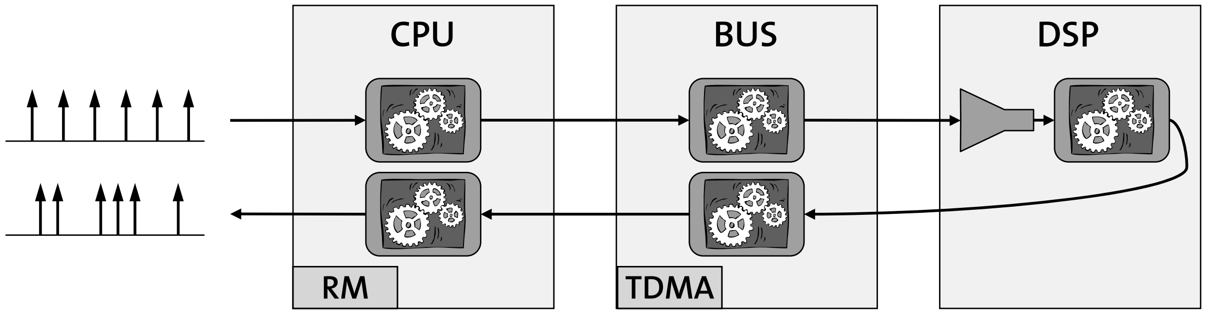
- Delays incoming events such that the output conforms to a given traffic specification.
- Guarantees that no events get delayed any longer than necessary.

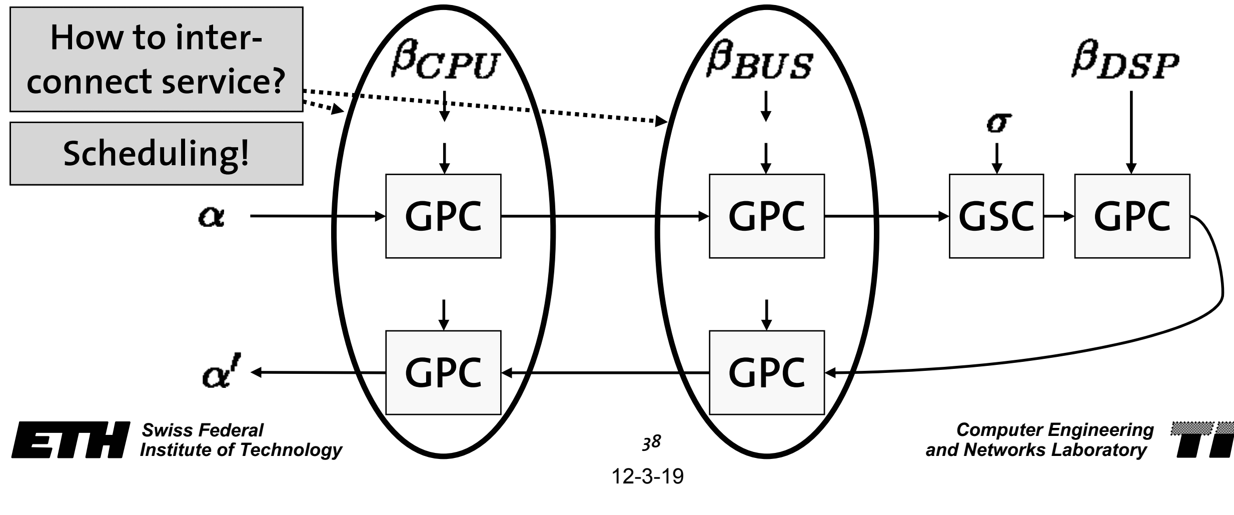
Real-Time Calculus

$$\alpha'^{u} = \alpha^{u} \otimes \sigma$$
$$\alpha'^{u} = \alpha^{l} \otimes (\sigma \overline{\oslash} \sigma)$$

System Composition



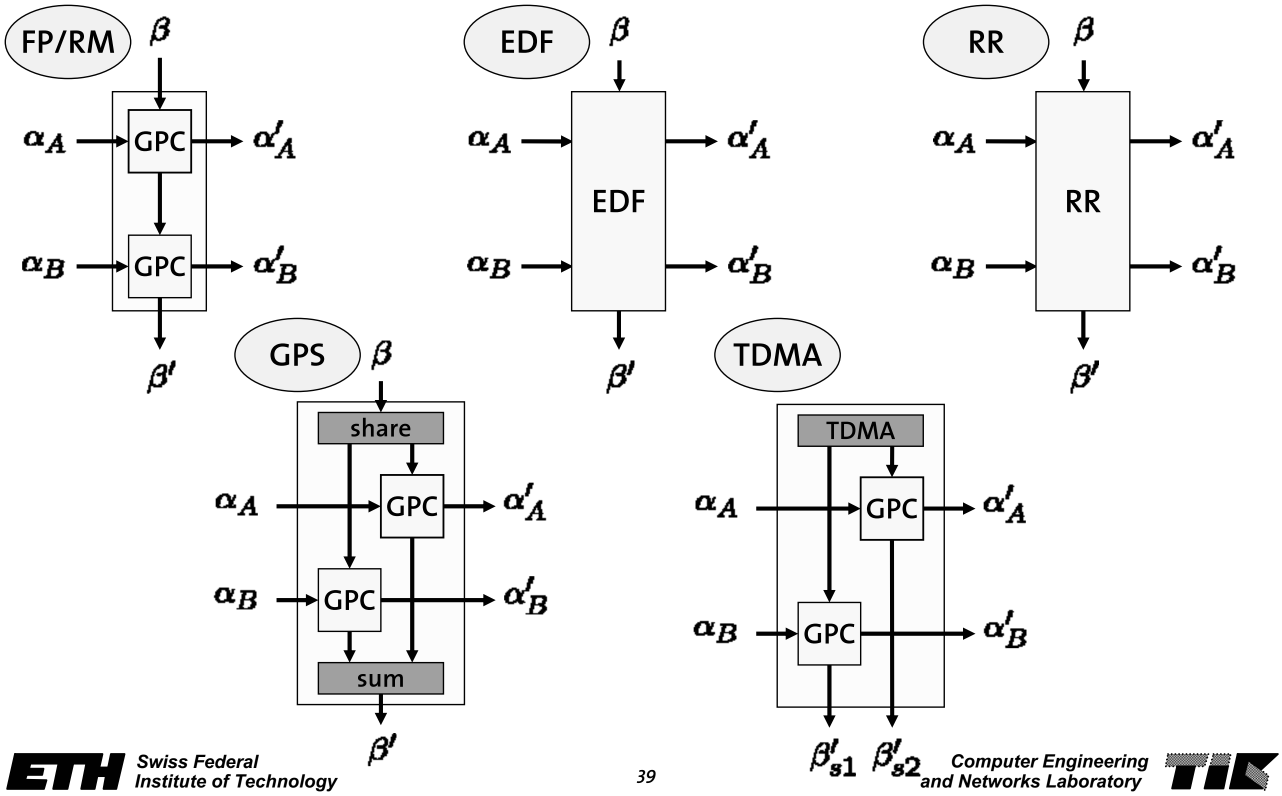




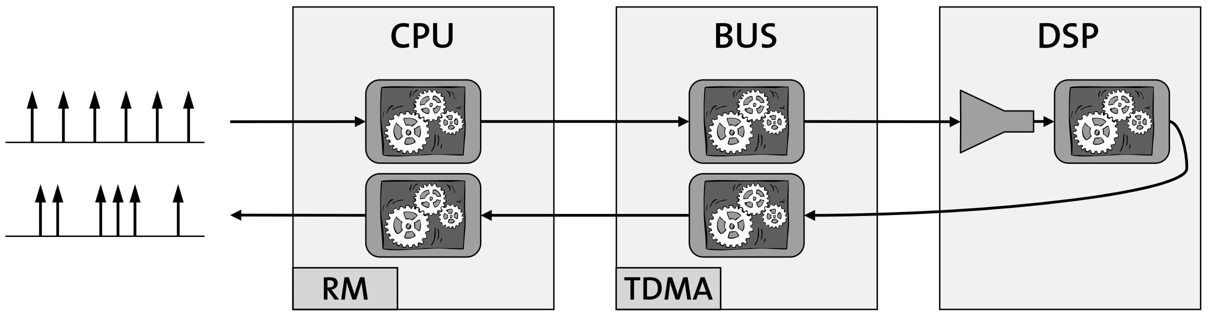
How to inter-connect service?

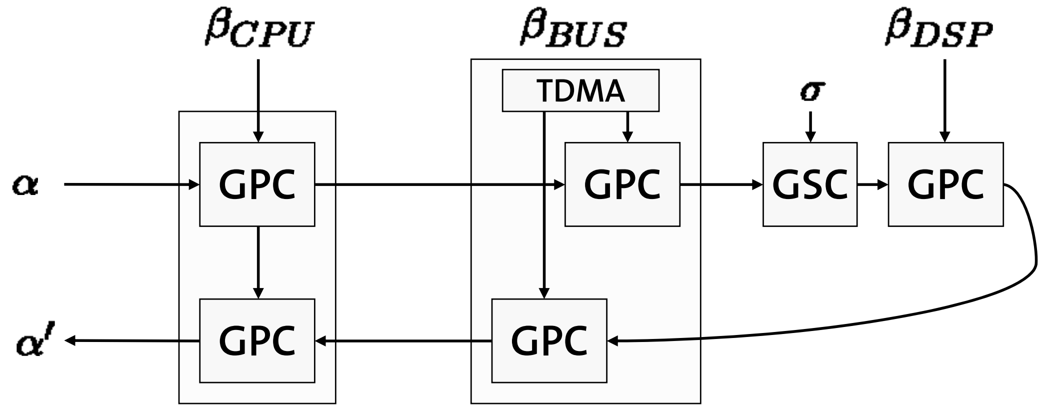
Scheduling!

Scheduling and Arbitration

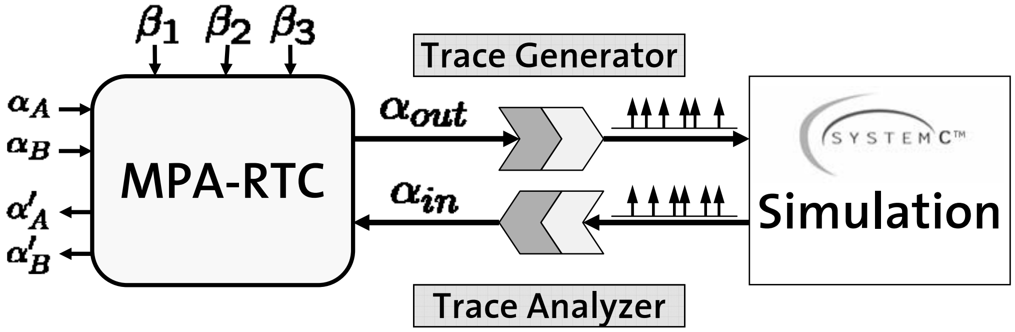


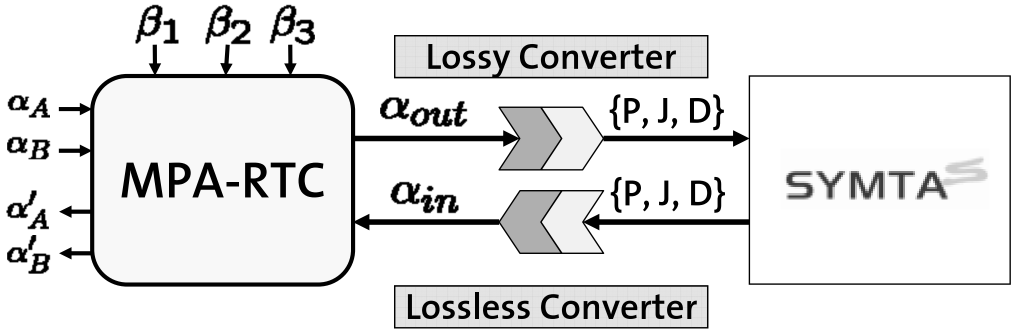
Complete System Composition





Embedding into other Frameworks





Applications

- *Interface-Based Design of Embedded Systems*
  – Check of Requirements at Composition-Time
  – Stepwise Refinement
  – Answering of design questions, e.g. resource dimensioning
- ***On-Line*** Load and Requirements ***Adaptation***
- ***Extensions***: activation schemes, processor state (cache), resource sharing (EDF, TDMA, Round Robin, Shapers), event types, blocking times.

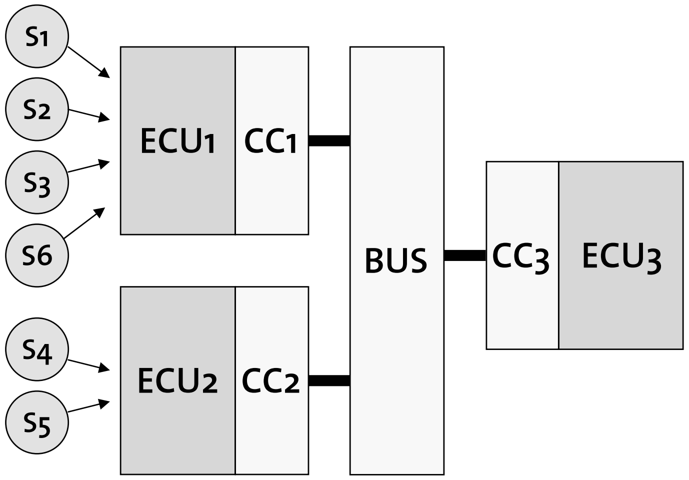
Experience

- ***Network processor modeling***
  - Detailed study of a network processor
  - Match between simulation and analytic methods
- Use in ***projects and case studies***
  - BridgeCo
  - Siemens
  - Netmodule
  - Embedded Systems Institute
- Integration into ***design space exploration***

Outline

- Embedding of Performance Analysis
- Modular Performance Analysis
- *Examples*

Case Study



6 Real-Time Input Streams
- with jitter
- with bursts
- deadline > period

3 ECU's with own CC's

13 Tasks & 7 Messages
- with different WCED

2 Scheduling Policies
- Earliest Deadline First (ECU's)
- Fixed Priority (ECU's & CC's)

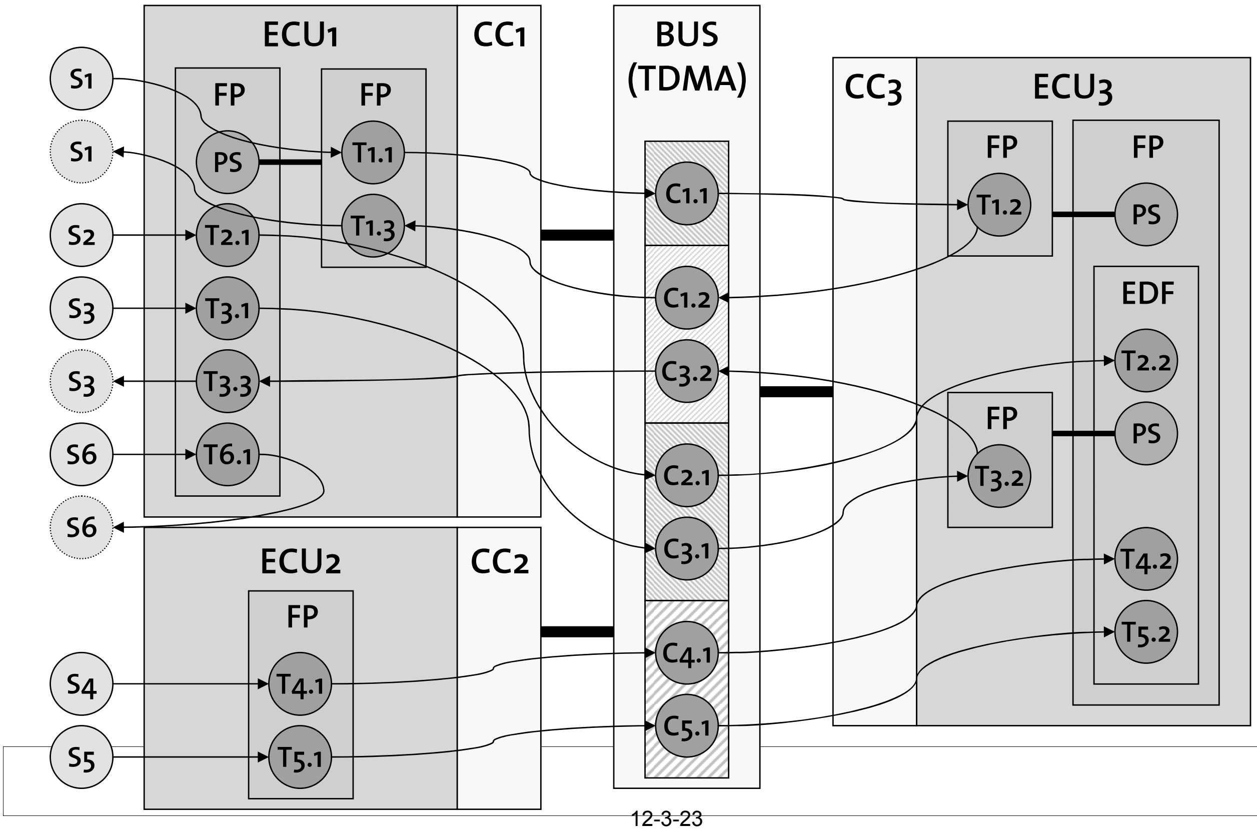
Hierarchical Scheduling
- Static & Dynamic Polling Servers

Bus with TDMA
- 4 time slots with different lengths (#1,#3 for CC1, #2 for CC3, #4 for CC3)

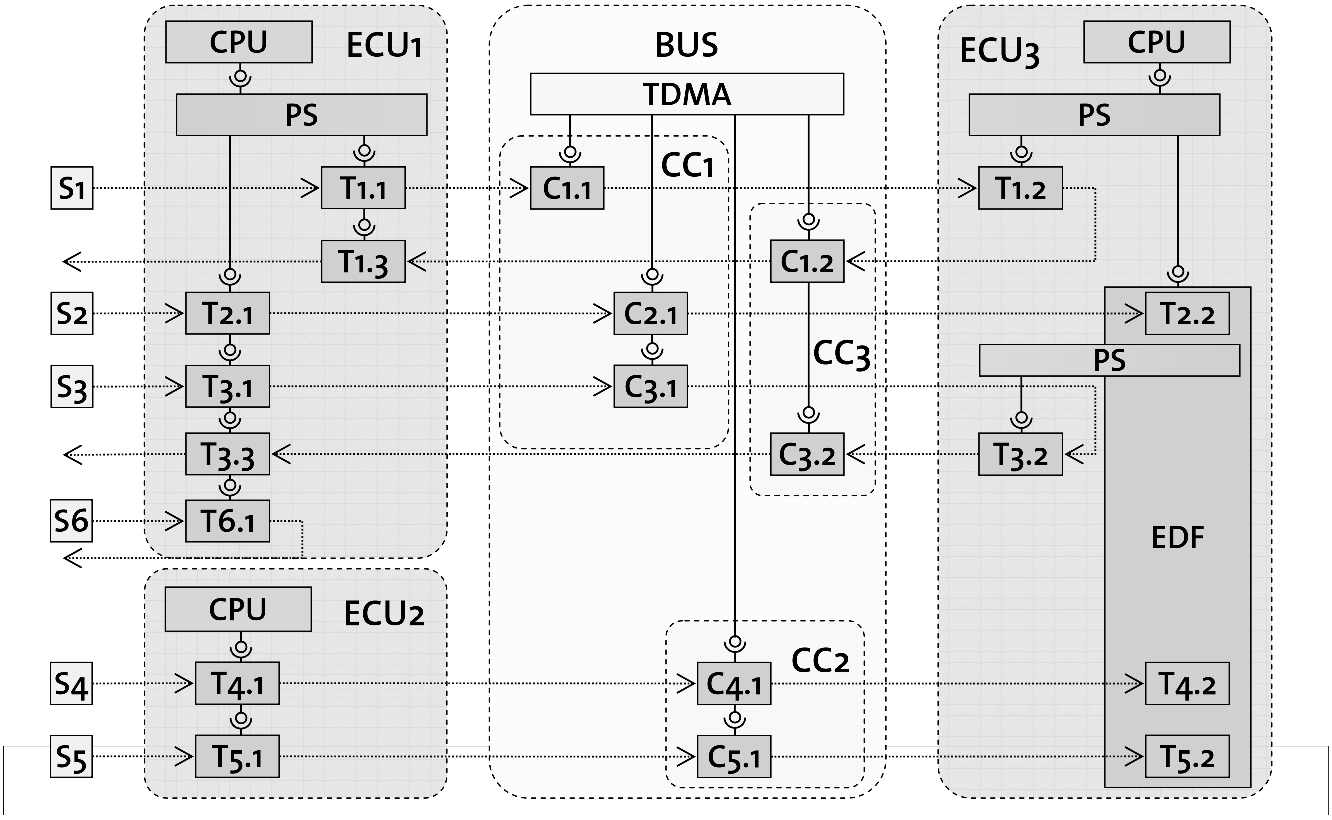
Total Utilization:

- ECU1	59 %
- ECU2	87 %
- ECU3	67 %
- BUS	56 %

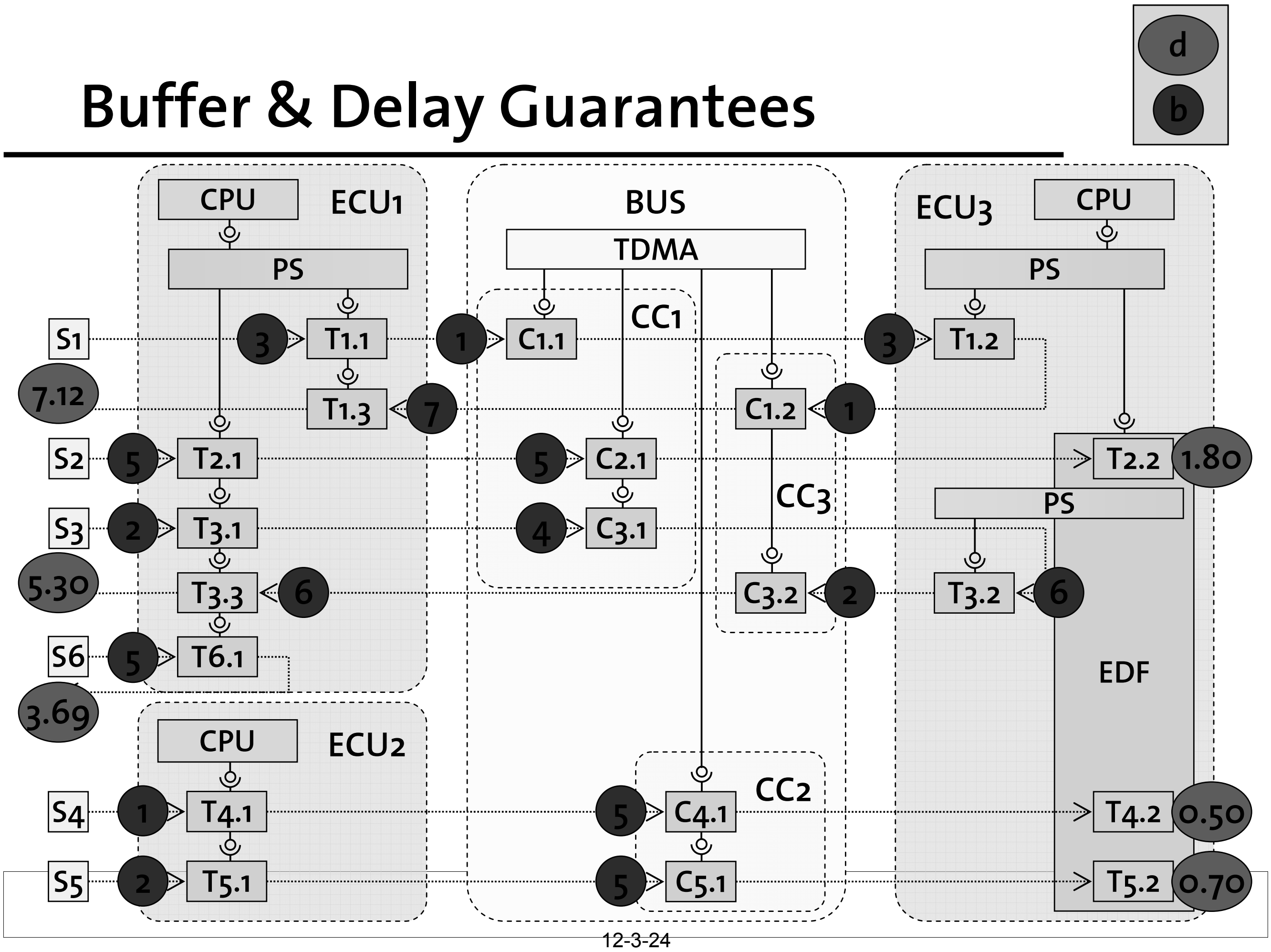
The Distributed Embedded System...



... and its Abstract Component Model

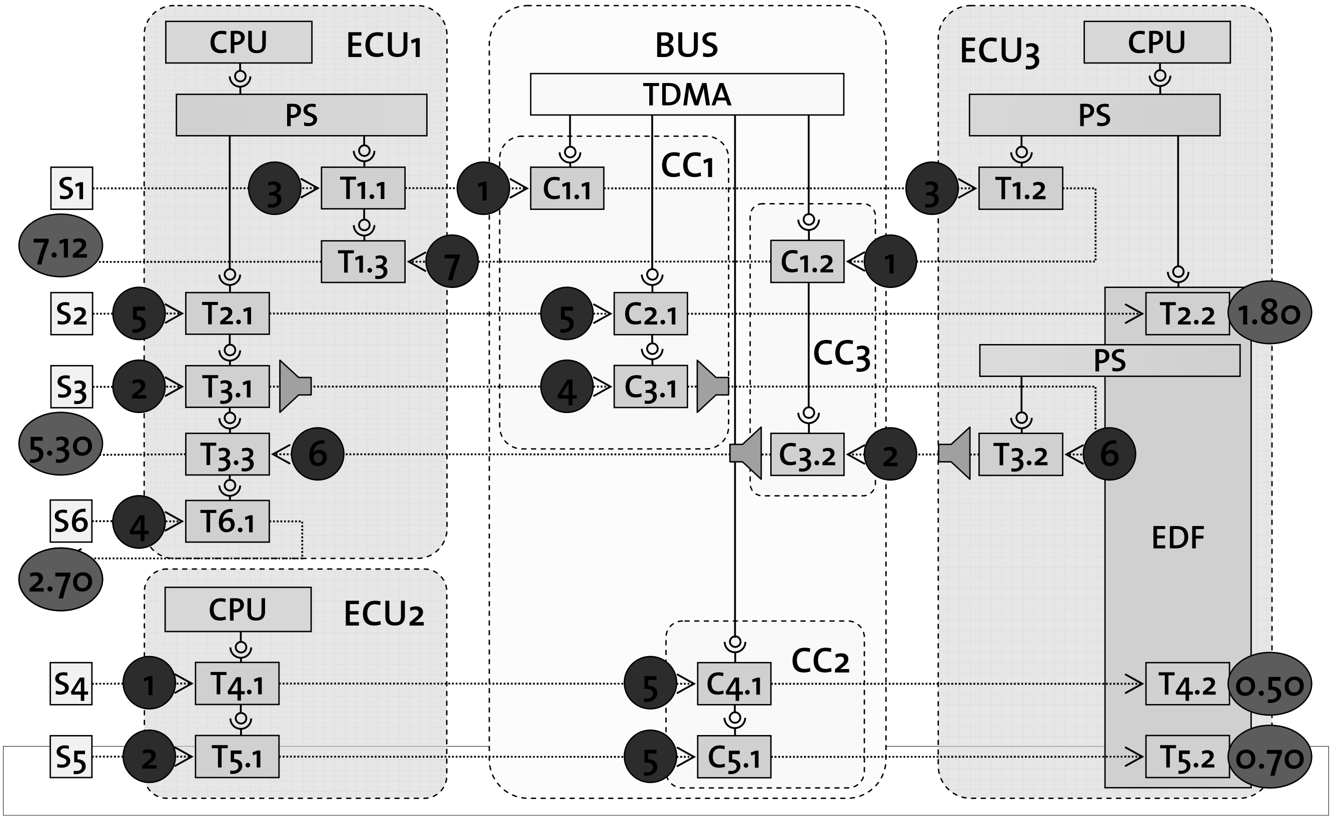


Buffer & Delay Guarantees

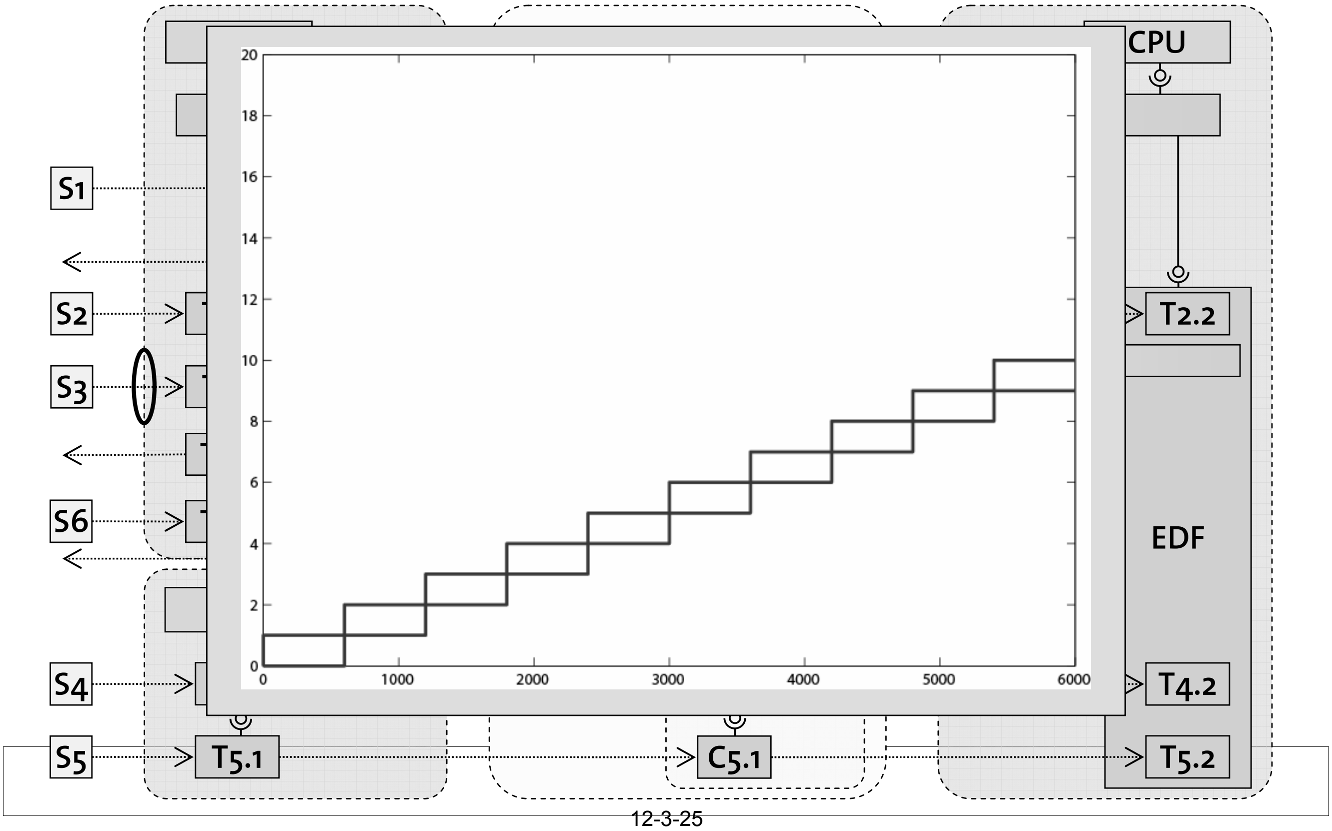


Adding Greedy Shapers

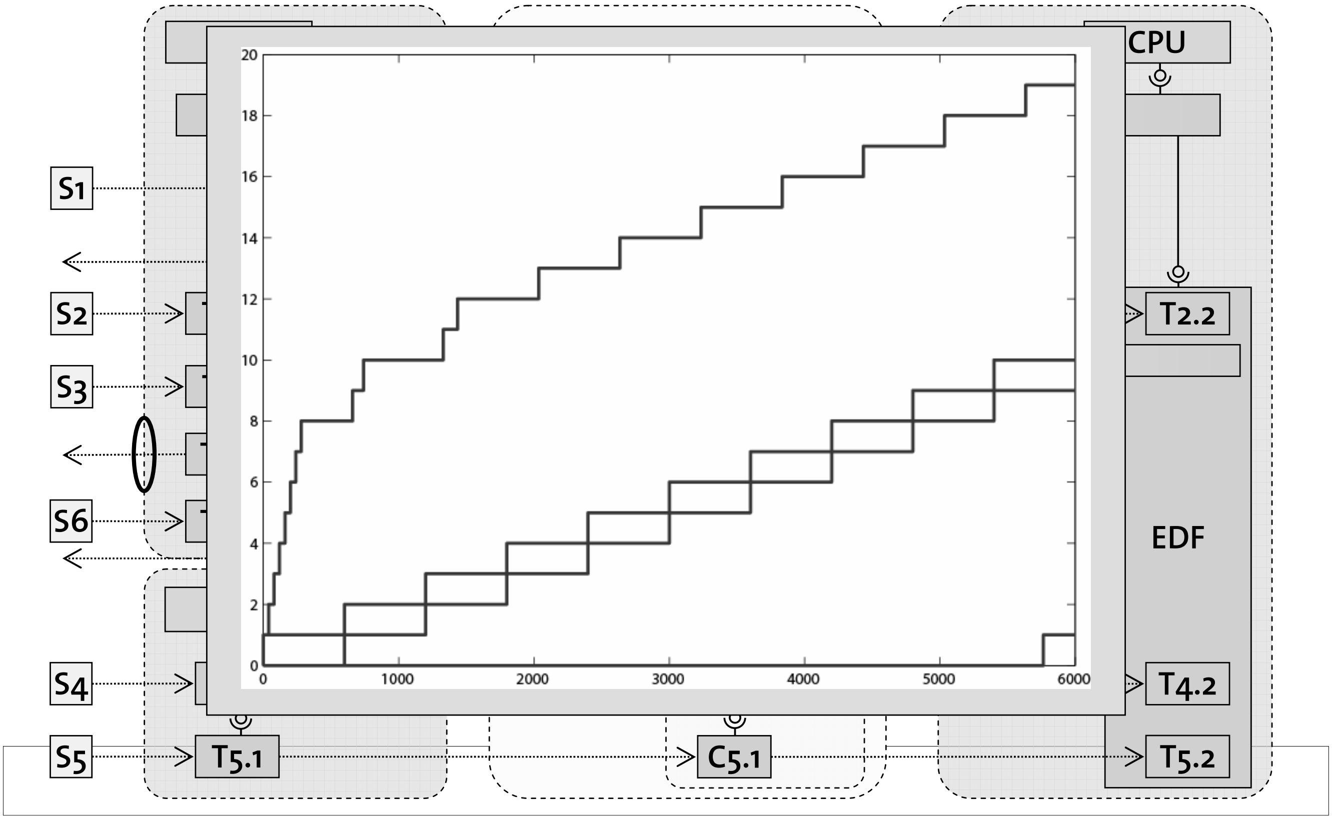
Delay D_{S6}: - 27%
Buffer B_{S6}: - 20%



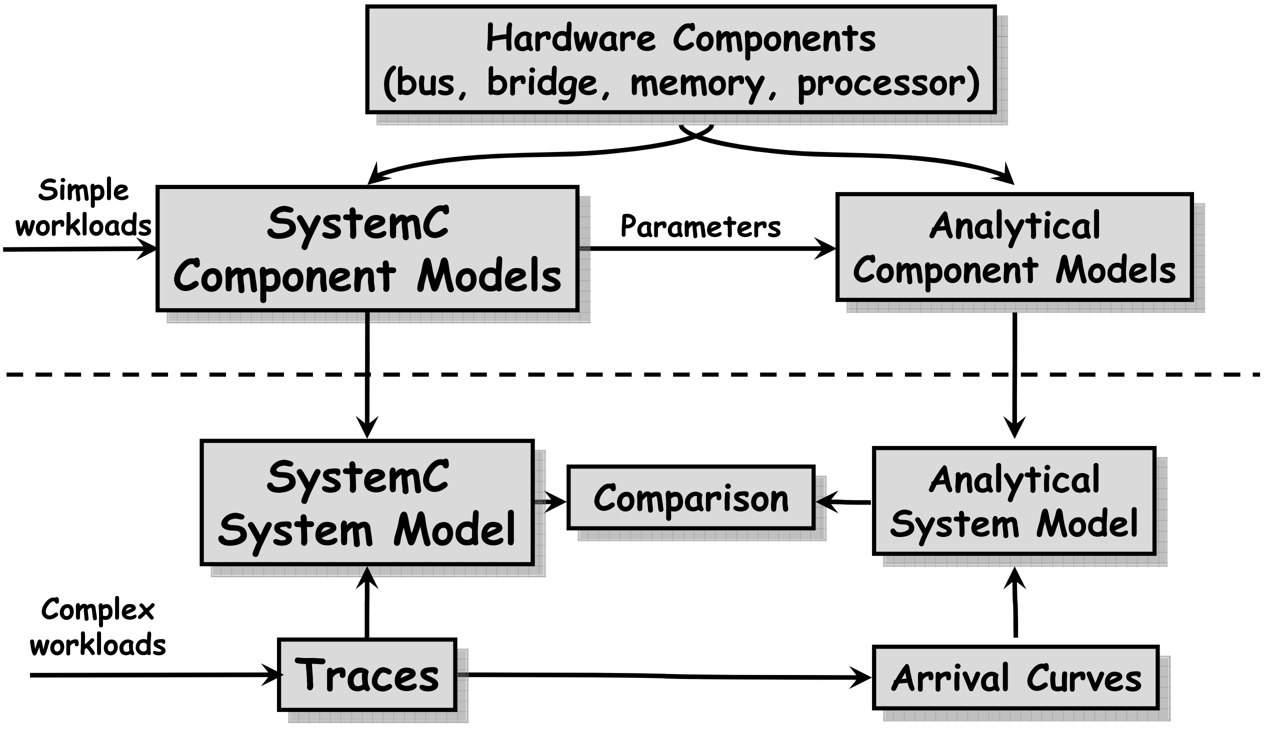
Input of Stream 3



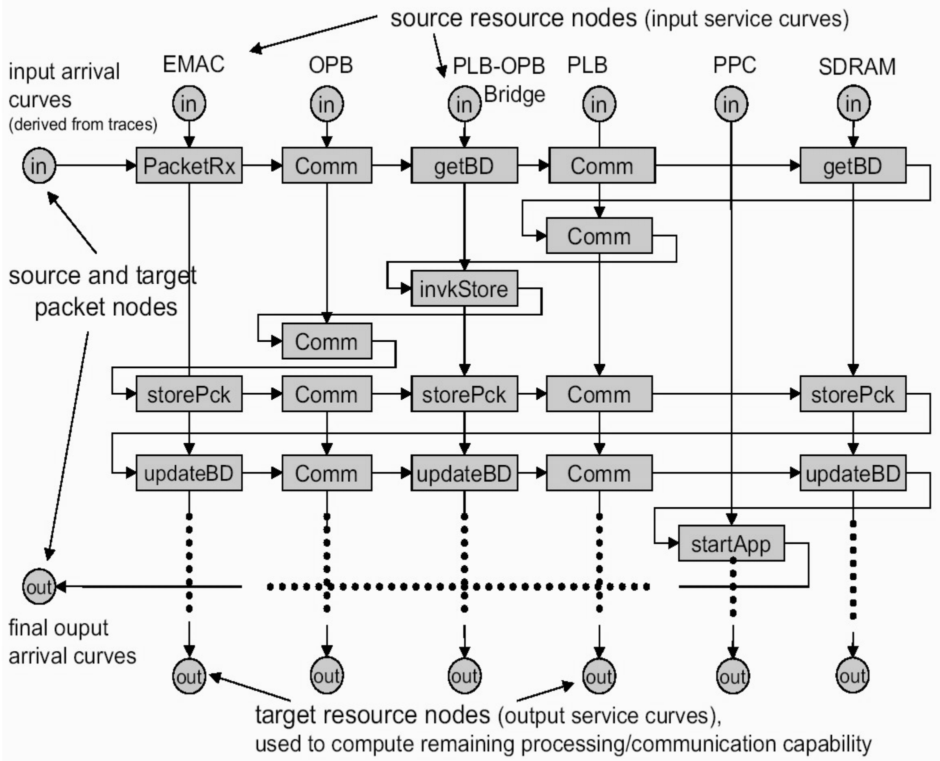
Output of Stream 3



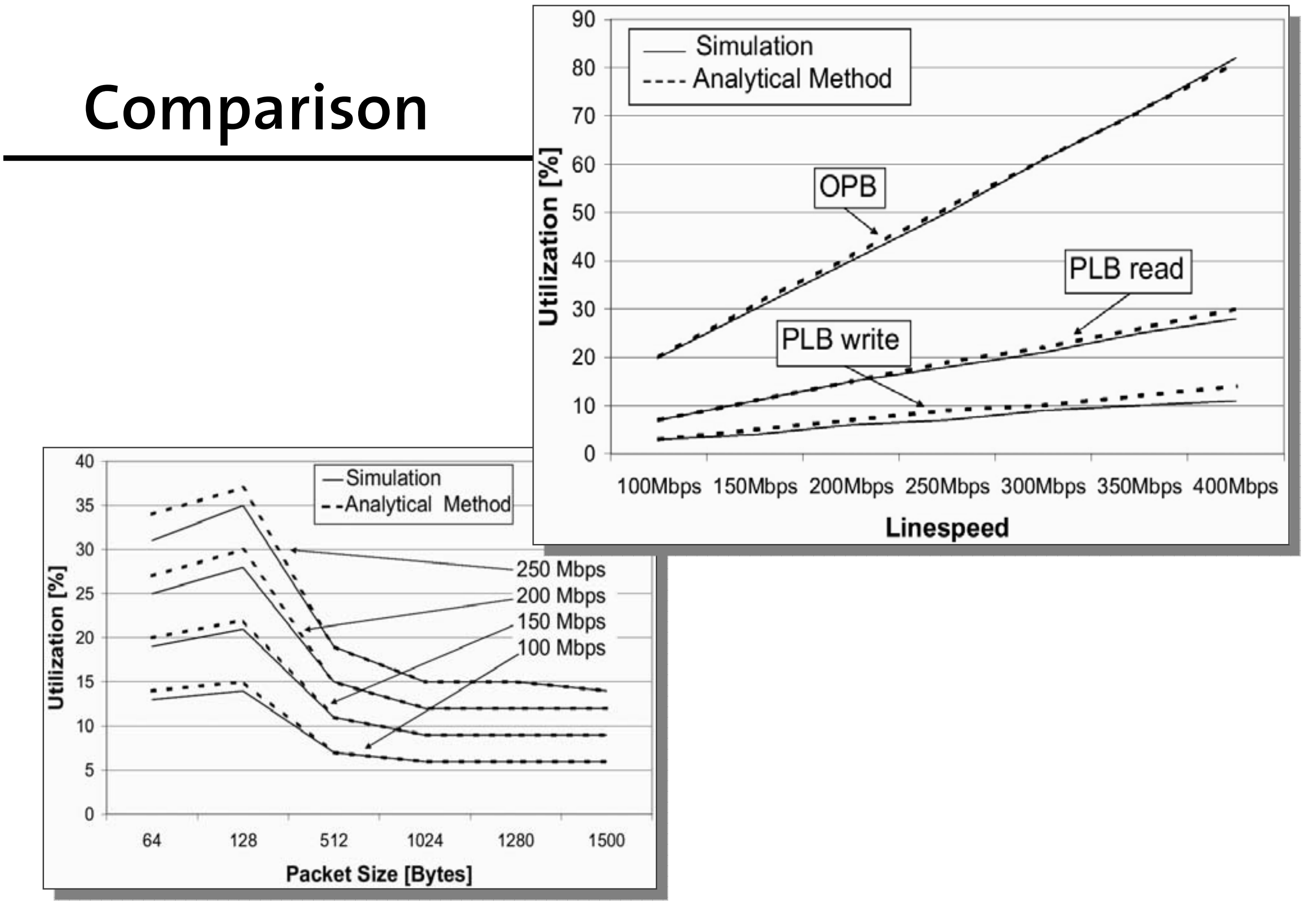
Does it Match Reality ? (IBM)



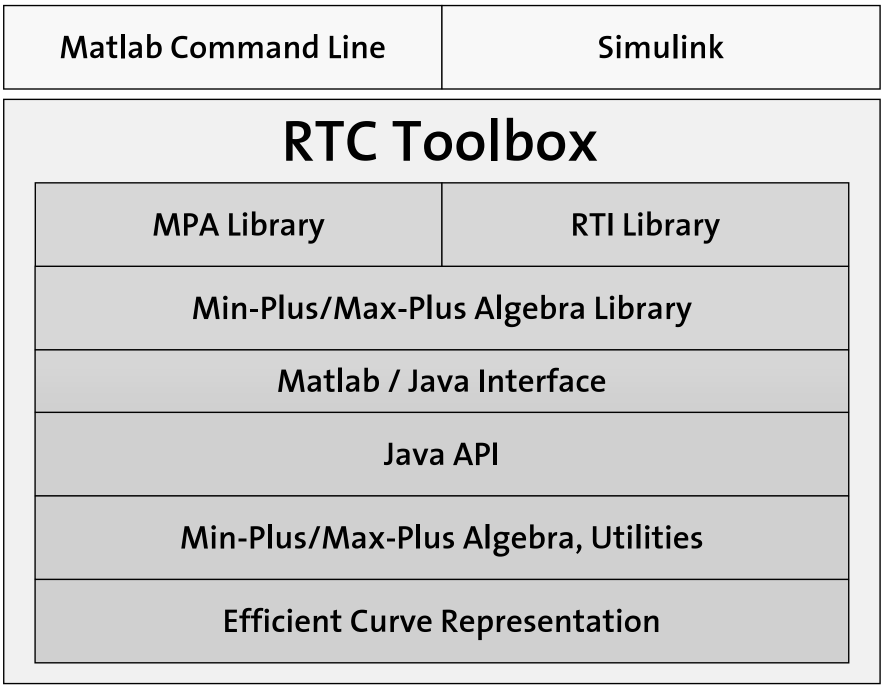
Network Processing Device


source resource nodes (input service curves)
input arrival curves (derived from traces)
EMAC
OPB
PLB-OPB Bridge
PLB
PPC
SDRAM
in
PacketRx
Comm
getBD
Comm
getBD
Comm
invkStore
Comm
storePck
Comm
storePck
Comm
storePck
updateBD
Comm
updateBD
Comm
updateBD
startApp
source and target packet nodes
out
final ouput arrival curves
out
target resource nodes (output service curves),
used to compute remaining processing/communication capability

Comparison



RTC Toolbox



Matlab Command Line | Simulink

RTC Toolbox

MPA Library | RTI Library

Min-Plus/Max-Plus Algebra Library

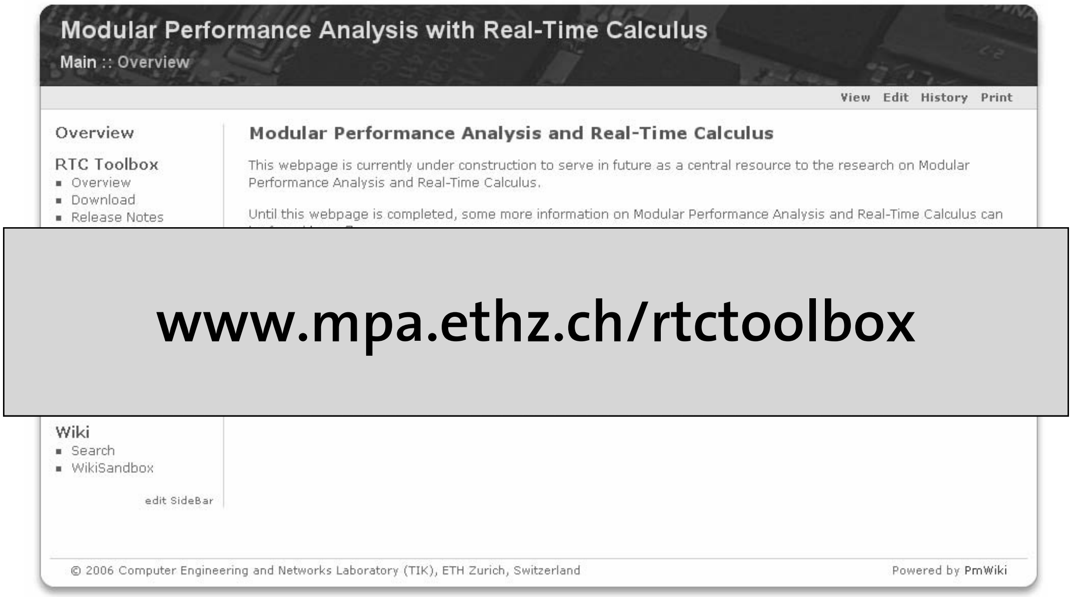
Matlab / Java Interface

Java API

Min-Plus/Max-Plus Algebra, Utilities

Efficient Curve Representation

RTC Toolbox: Version 0.9 Released



Modular Performance Analysis with Real-Time Calculus

Overview

RTC Toolbox
- Overview
- Download
- Release Notes

Wiki
- Search
- WikiSandbox

Modular Performance Analysis and Real-Time Calculus

This webpage is currently under construction to serve in future as a central resource to the research on Modular Performance Analysis and Real-Time Calculus.

Until this webpage is completed, some more information on Modular Performance Analysis and Real-Time Calculus can

© 2006 Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Switzerland

Powered by PmWiki

www.mpa.ethz.ch/rtctoolbox

Acknowledgement

- Collaborators:
  - Ernesto Wandeler
  - Samarjit Chakraborty
  - Simon Künzli
  - Alexander Maxiaguine
  - Nikolay Stoimenov
  - Simon Perathoner
- Funding:
  - SNF, KTI, MEDEA+/SPEAC, ARTIST2 NoE, EU IP SHAPES