


A Co-Simulation Framework for MPSoC Run-Time Behavior Analysis in Early System Design

Anastasia Stulova, Jianjiang Ceng, Jeronimo Castrillon, Weihua Sheng,
Rainer Leupers

MCC09

26 Nov 2009

Institute for Integrated Signal Processing Systems

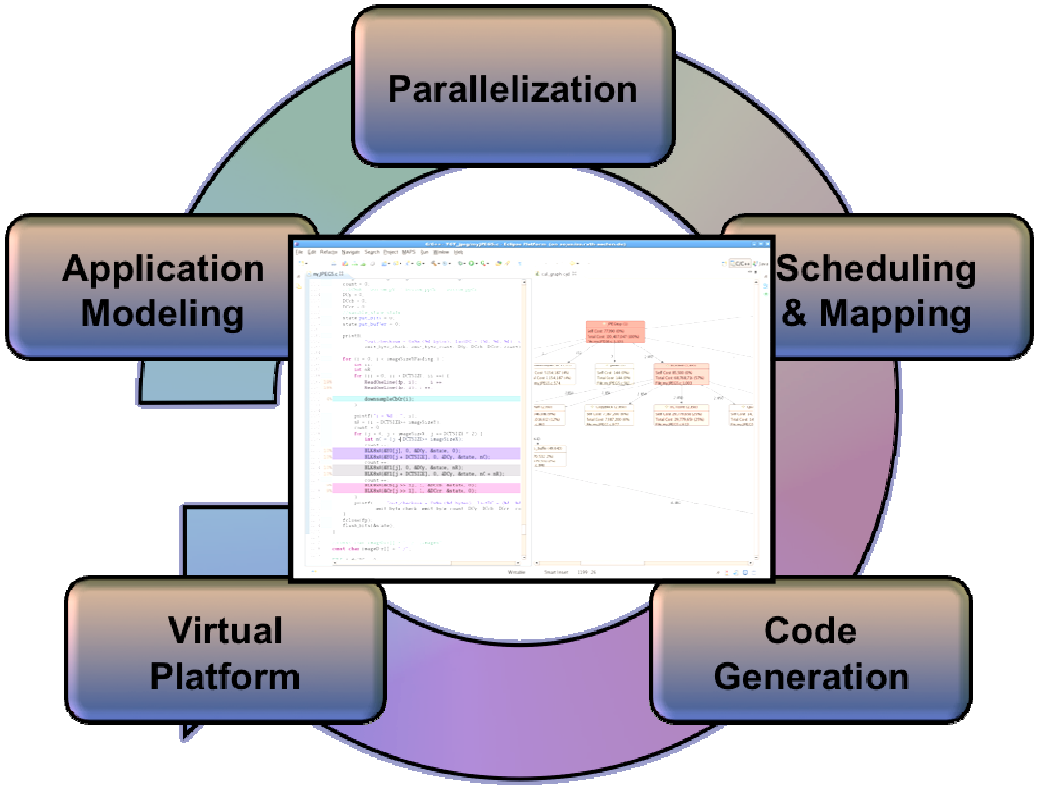
Outline

- **A Brief Introduction of the MAPS Project**
- **A High-level Virtual Platform (HVP)**

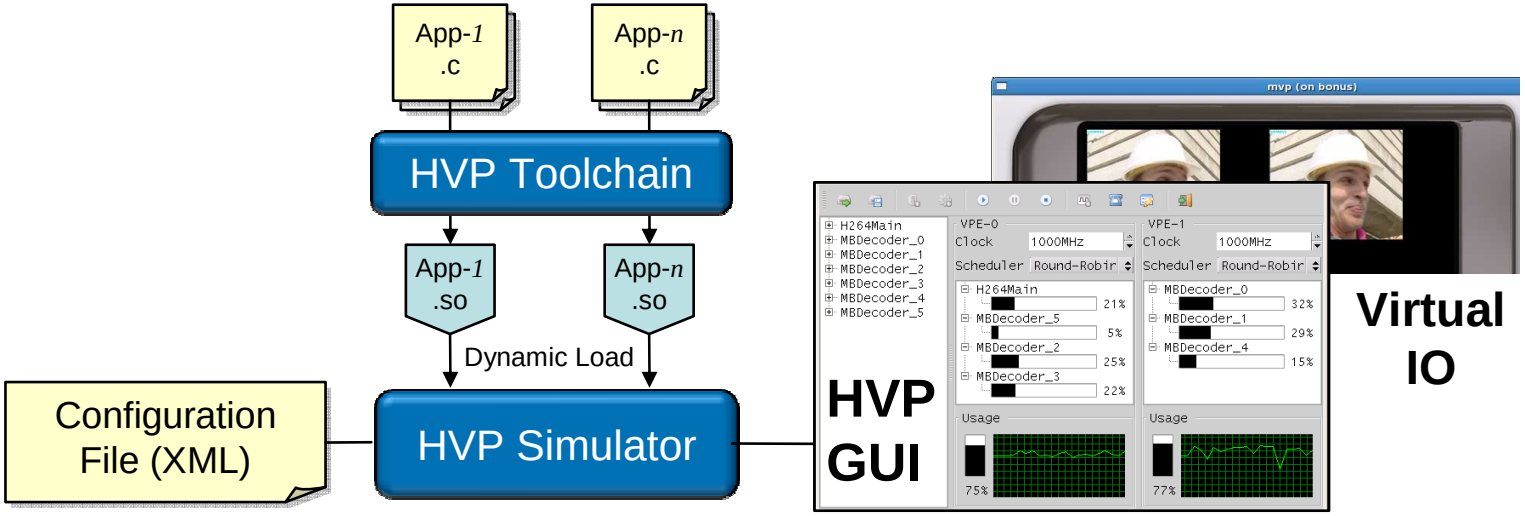
- **OS Modeling with HVP**
  - Motivation and Related Work
  - A Virtual OS Model
  - Scheduler and Timing Model

- **A Tool Demonstration**
- **Conclusions and Future Work**

- **MAPS is a research project at RWTH Aachen University which targets at the problem of MPSoC software development**



A High-Level Virtual Platform (HVP)

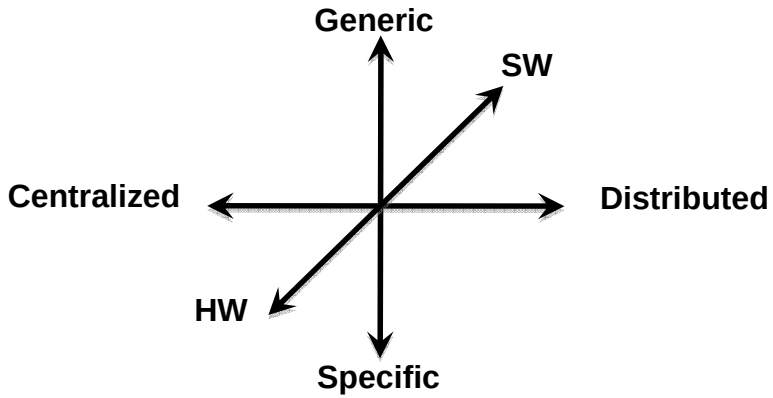


- **C applications are directly compiled by the HVP specific toolchain into dynamic libraries**
  - determines virtual execution time
- **The SystemC based HVP simulator dynamically loads the compiled binaries**
- **Configurations can be stored as XML files for reconfiguration**
- **A virtual IO device is available, which can be used to visualize the execution result directly**

Role of OS in ESL

- **Why simulating OS in Early System Design?**
  - It changes behavior of the system by determining execution order of tasks
  - It contributes with the execution time latency
- **Embedded OS/RTOS in Multi-Core Era**

© **A Quick Safari Through the MPSoC Run-Time Management Jungle,** Vincent Nollet, Diederik Verkest, Henk Corporaal, **Journal of Signal Processing Systems, Springer New York, November 2008**



- **Two major design flows:**
  - Selection from existing products
  - Synthesis (MAPS, SHAPES)

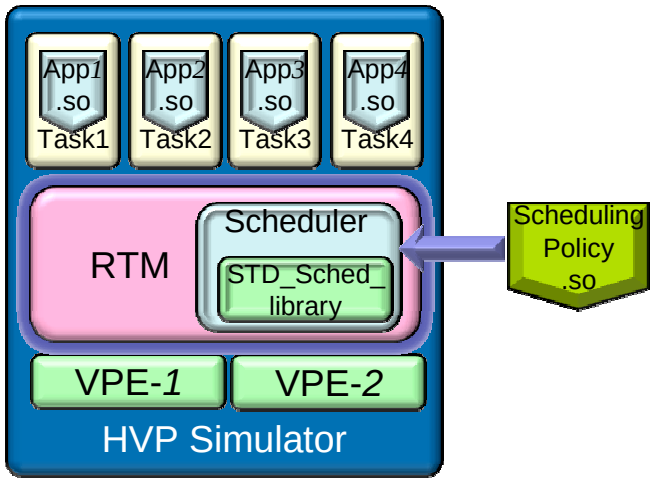
Related Works

- **OS model as an SLDL extension:**
  - A. Gerstlauer et al, F. Hessel et al, Y. Yi et al
- **Generic OS in tools:**
  - VPU Coware, MESH, R. Le Moigne CoFluent Studio™

- **Main limitations:**
  - Centralized and migrative scheduling schemes are not supported by most frameworks
  - Retargetability is poorly addressed
  - No statistics dedicated to the OS designer

A Virtual OS Model

- **Modeled components:**
  - Task Management
  - Task Scheduling
- **Represented by:**
  - APIs to the application designer:
    - Tasks Communication, Synchronization, Scheduling
  - SystemC Simulator:



- *VPE – Virtual Processing Element*
- *RTM - Run-Time Manager*

OS Scheduler

- **Scheduling event – a change in the simulator which might cause rescheduling of tasks in the system:**
  - Ex: task state transition; change of VPE *param*; end of a time slice; task arrival/termination;

    ➔ *Determines a type of the event*
- **Scheduling function:**
  - *update_sched(sched_evnt) → (sched_matr, ts_vect)*
  - *sched_evnt is a touple (time_stamp, evnt_type,* $Task_i$*,* VPE_i*)*
  - *sched_matr is a* N_T *x* N_{VPE} *matrix* (determines task dispatching to VPEs)
    - N_T *– the number of tasks in the system;*
    - N_{VPE} *– the number of VPEs in the platform;*
  - *ts_vect – a vector with time slice values for the next task activations (size* N_T*)*

  ➔ *Called by RTM every time scheduling event arrives*

  ➔ *Fully customizable by the user*

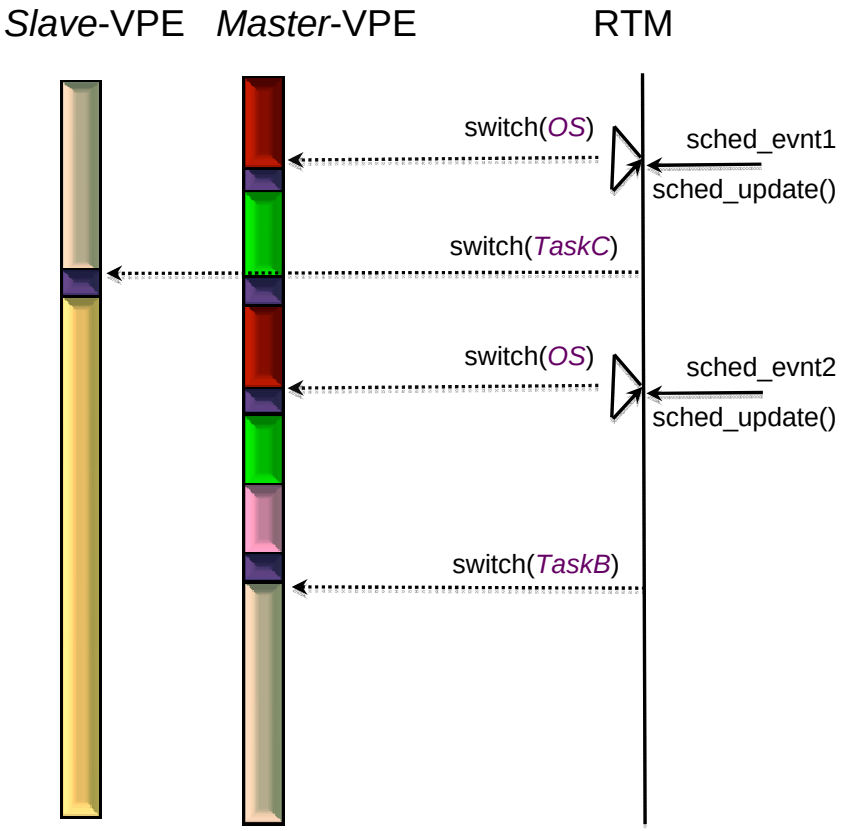
OS Timing Model

- **Save/Load task context time:**
  - Parameterizable via a configuration file
- **Scheduling Latency:**
  - Delay annotation *HVPConsumeTime(int cycles);*
  - Scheduling code instrumentation

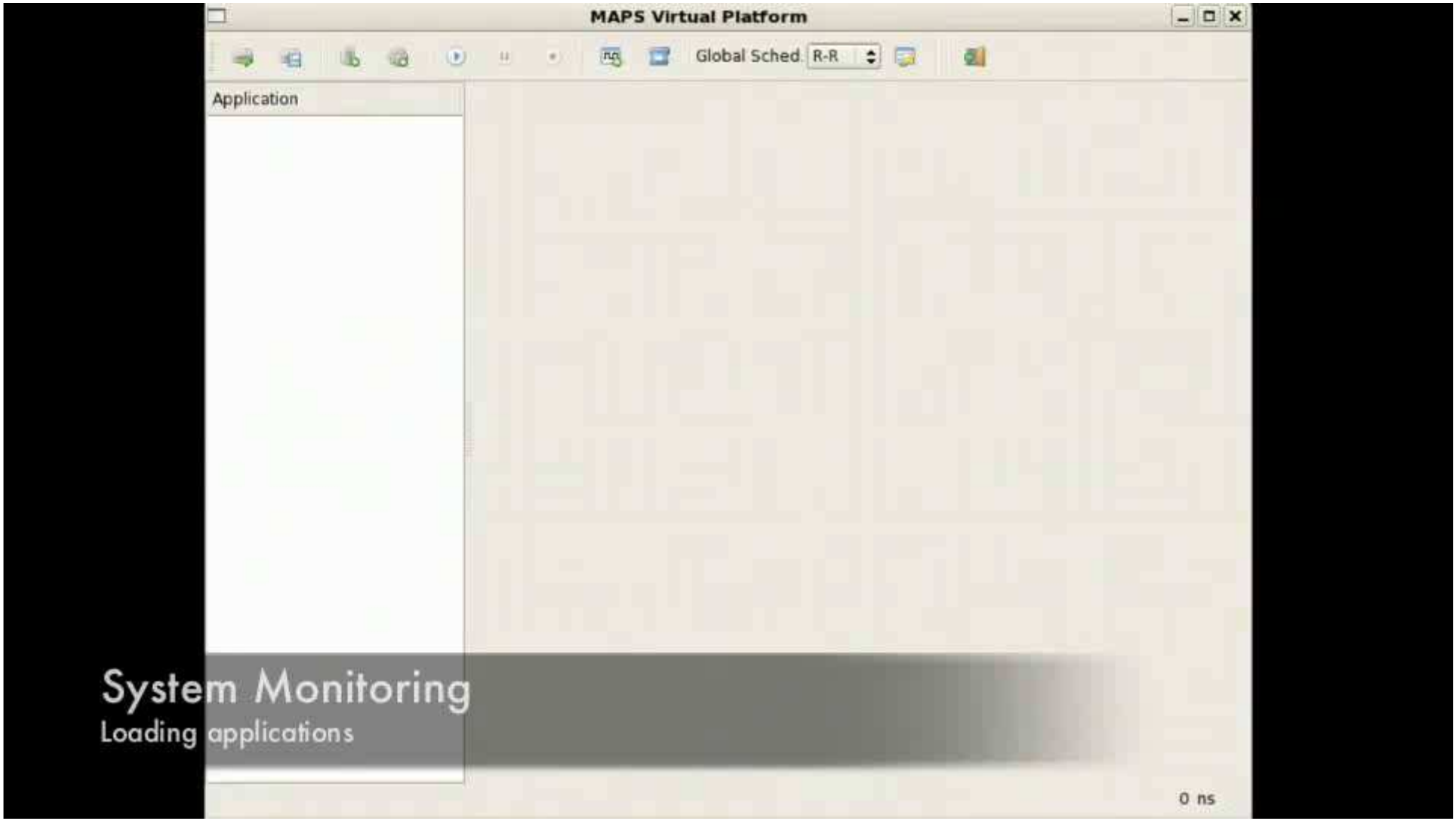
    -> counts virtual # cycles scheduling code takes (as HVP toolchain)
- **Migration Penalty:**
  - Parameterized by a constant value (one value per each pair of processors and a task)
  - Considered only if a task migrates b/w VPEs

Example of Centralized Control

TaskA
TaskB
TaskC



SW Demo

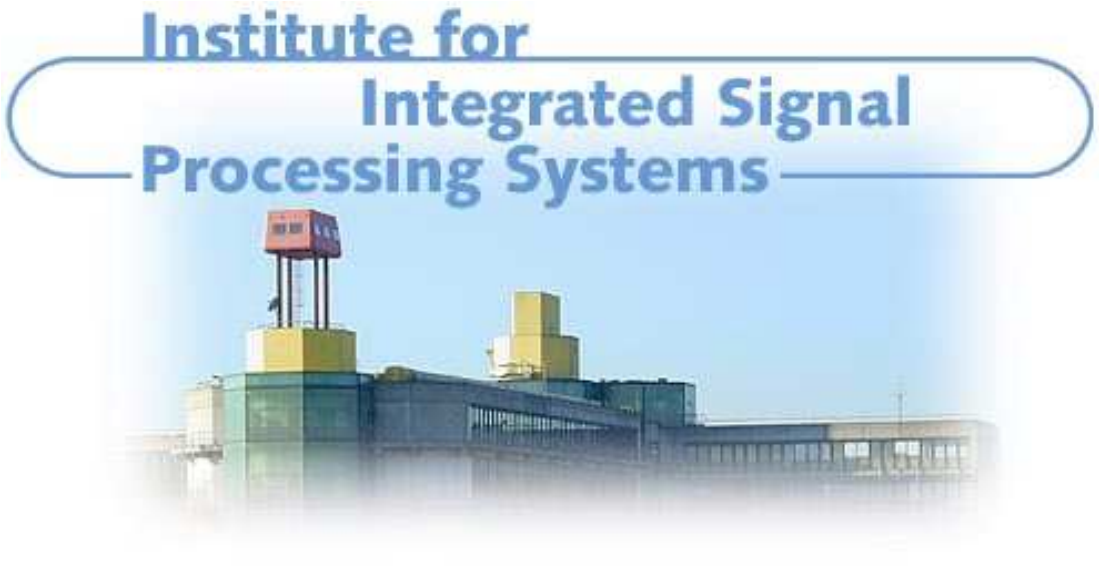

MAPS Virtual Platform
Global Sched.
R-R
Application
System Monitoring
Loading applications
0 ns

Conclusions

- **A generic OS model which enables to model a broad range of run-time management schemes was presented**
- **OS scheduling type and timing behavior can be easily changed**
- **The OS model was successfully integrated into the HVP tool**
- **We provide various “on-the-fly” and post execution information about system behavior to be analyzed by OS and application developers**

Future Work

- **Evaluate the accuracy of the trend prediction w.r.t ISS Virtual Platforms and real HW**
- **Facilitate a more accurate model of the task migration penalty (including automatic migration time computation)**



Thank you !