SILEXICA

Towards a flexible compiler infrastructure for heterogeneous MPSoCs: Two case studies

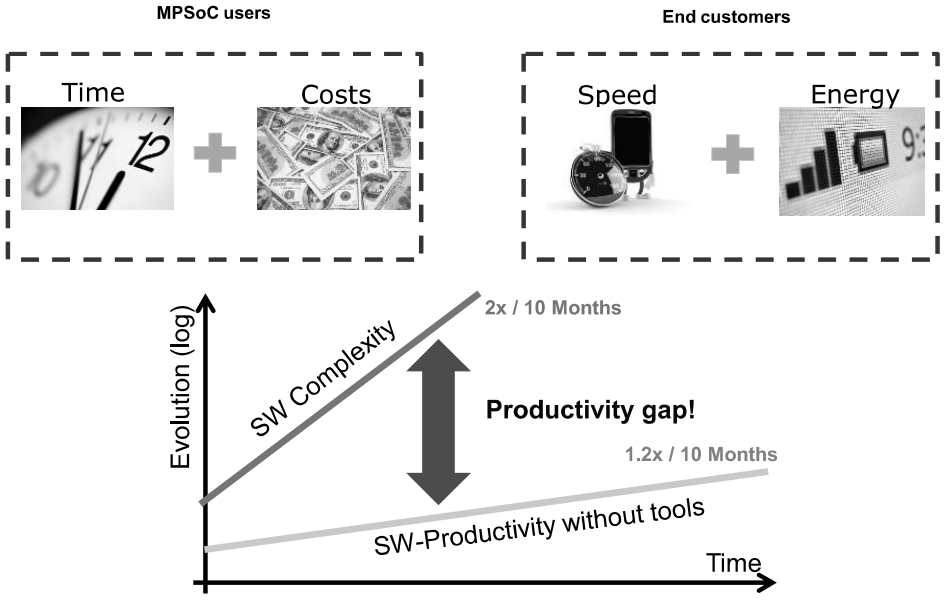
Weihua Sheng

MPSoC Forum, Ventura Beach, 2015

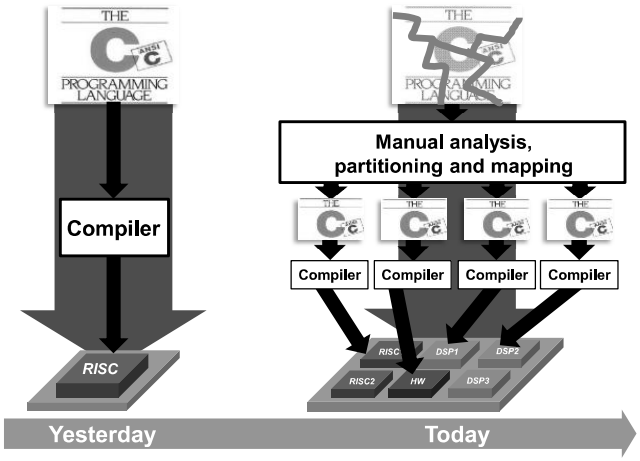
About Silexica

- **Spin-off (2014) from ICE institute at RWTH Aachen University**
- **Commercializing RWTH´s MAPS compiler technology for embedded multicore architectures**
- **Programming tools portfolio:**
  - **SLX Parallelizer - automated parallelization of sequential legacy C code**
  - **SLX Mapper – parallel task-to-processor software distribution with extensive analysis facilities**
  - **SLX Generator – automatic native target C code generation, including inter-task communication code**
  - **SLX Explorer – fast early multicore hardware platform selection and joint software optimization**
- **Currently addressed application domains:**
  - **Mobile devices**
  - **Wireless communication**
  - **Automotive electronics**

MPSoC software development: status quo

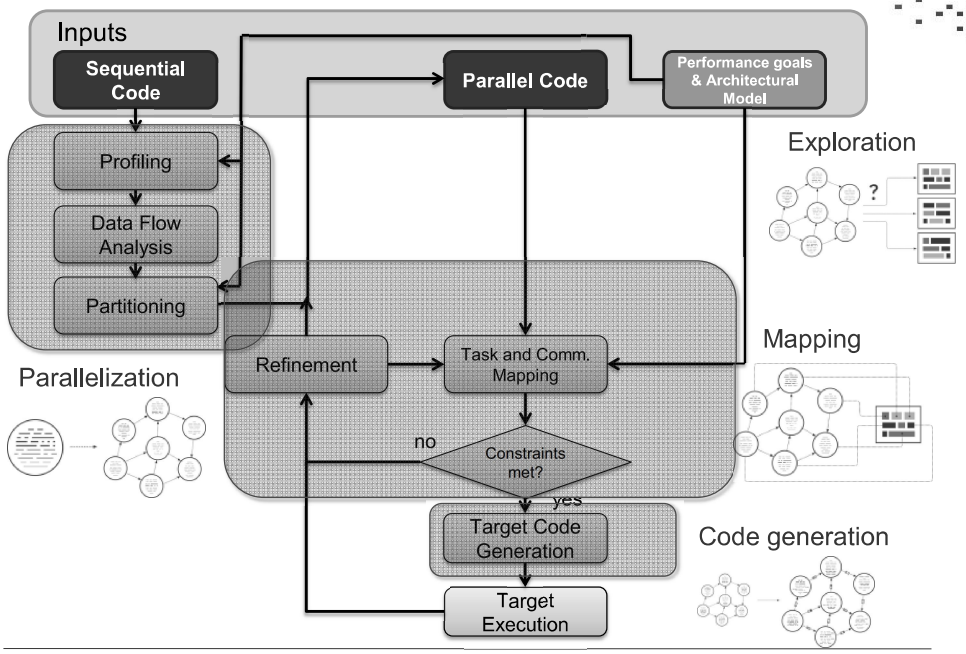


Blackbox compiler does not scale for MPSoC



- Compiler is not able to be kept in **blackbox** and executed fully **automatically** anymore.
- Compiler is still wanted to improve SW development productivity
  - Demand a **flexible compiler infrastucture** that can handle MPSoC complexity and various user requirements

A MPSoC compiler example from user's view

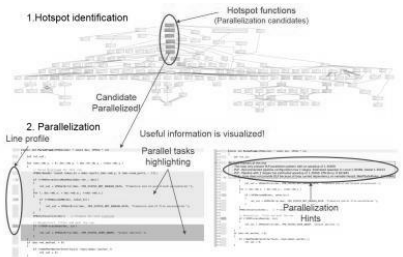


Case study 1

Sub-optimal usage of hardware of existing software stack

- Situation
  - Android phone manufacturer
  - Core libraries (Image encoding/decoding) are not multicore-ready
  - Concerns about performance and energy consumption from parallelization

- Infrastructure applicablity
  - ✓ **Parallelization**
  - ✓ **Code generation**



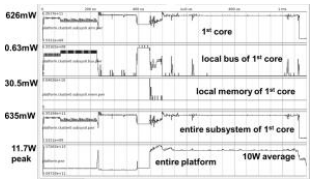
- Special requirements
  - On target profiling
  - Mixed language SW stack (Java/C/C++)

- ✓ 17% less energy consumption
- ✓ 38% performance increase
- ✓ Faster end-user applications

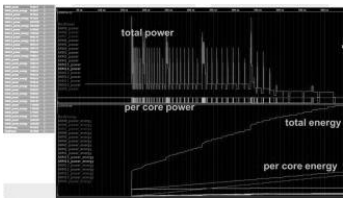
Case study 2

Which multicore platform is optimal for a new LTE base station?

- Situation
  - Wireless infrastructure provider
  - Power-aware multicore SW-to-processor mapping
  - Concerns about power (a) vs different mapping (b) vs different scenarios



- Infrastructure applicablity
  - ✓ **Mapping**
  - ✓ **Exploration**



- Special requirements
  - Target platform power model
  - Virtual testing platform

- ✓ Huge optimization opportunity for average and peak power
- ✓ Trade-off of runtime vs power for multicore systems
- ✓ Optimal Platform can be selected in a short time

Summary

- SW will be THE differentiator for future embedded products
- A compiler infrastructure for heterogeneous MPSoCs should be flexible
  - to be customized for different usage scenarios
  - to incorporate / collaborate with other design tools
  - to support retargeting for different architectures

Thank you! Questions?

SILEXICA
multicore meets simplicity