New Research Group: System Software at Friedrich Schiller University Jena

Clemens Grelck

Herbst-Treffen der GI-Fachgruppe Betriebssysteme
Bamberg, September 2023

Some words about me ...

© Friedrich-Schiller-Universität

Professor for system software:

- Started April 2023
- Research group in the making...
- We're hiring ...

System software:

- Bridging the gap between hardware and application software
- Operating systems
- Programming languages
- Compilers
- Runtime systems
- ...

Friedrich-Schiller-University Jena:

- Founded in 1564
- 17,000 students
- 16 professors in computer science
- Only full university in Thüringen

Some words about me ...

But how did I get here?

© Friedrich-Schiller-Universität

Christian-Albrechts-Universität zu Kiel

1990–2001

- Student of informatics and economics
- Diploma in informatics
- Wissenschaftlicher Angestellter
- Dr.rer.nat

Universität zu Lübeck

© Christian Wolf (www.c-w-design.de), CC BY-SA 3.0

© K.ristof, CC BY-SA 3.0

2001–2008

- Wissenschaftlicher Angestellter
- Wissenschaftlicher Assistent (C1)

University of Hertfordshire, Hatfield, United Kingdom

© Allan Engelhardt, CC BY-SA 2.0

© Hatfield Park

2006–2009

- Research Fellow
- Principal Lecturer

Universiteit van Amsterdam, Niederlande

© Massimo Catarinella, CC BY-SA 3.0

© Nikolai Karaneschev, CC BY 3.0

2008–2023

- Assistant Professor
- Associate Professor

And finally ...

© Friedrich-Schiller-Universität

So much about me ...

But what about my research ?

My research agenda

System software to the rescue:

- Domain-specific programming languages and abstractions
- Optimising compilers / static analysis
- Adaptive runtime systems / dynamic analysis and control

... targeting (heterogeneous) parallel architectures

My research agenda

System software to the rescue:

- Domain-specific programming languages and abstractions
- Optimising compilers / static analysis
- Adaptive runtime systems / dynamic analysis and control

... targeting (heterogeneous) parallel architectures

With non-functional requirements:

- time: speed / deadline
- energy: saving / budget
- robustness / fault-tolerance
- ...

Programming Language SAC: Single Assignment C

Key design choices:

- Purely functional semantics with C-like syntax
- Data-parallel array programming
- Targeting compute-intensive applications
- Immutable, truly multi-dimensional arrays
- **Radically different: completely target- and resource-agnostic**

Programming Language SAC: Single Assignment C

Key design choices:

- Purely functional semantics with C-like syntax
- Data-parallel array programming
- Targeting compute-intensive applications
- Immutable, truly multi-dimensional arrays
- **Radically different: completely target- and resource-agnostic**

Zero intertwining with:

- memory management code
- parallelisation code
- synchronisation/communication code
- target hardware-specific arrangements
- ... and all the other clutter

Programming Language SAC: Single Assignment C

Key design choices:

- Purely functional semantics with C-like syntax
- Data-parallel array programming
- Targeting compute-intensive applications
- Immutable, truly multi-dimensional arrays
- **Radically different: completely target- and resource-agnostic**

Zero intertwining with:

- memory management code
- parallelisation code
- synchronisation/communication code
- target hardware-specific arrangements
- ... and all the other clutter

Benefits:

- No race conditions
- No space leaks
- No deadlocks
- No .... you name it

Programming Language SAC: Single Assignment C

Key design choices:

- Purely functional semantics with C-like syntax
- Data-parallel array programming
- Targeting compute-intensive applications
- Immutable, truly multi-dimensional arrays
- **Radically different: completely target- and resource-agnostic**

Zero intertwining with:

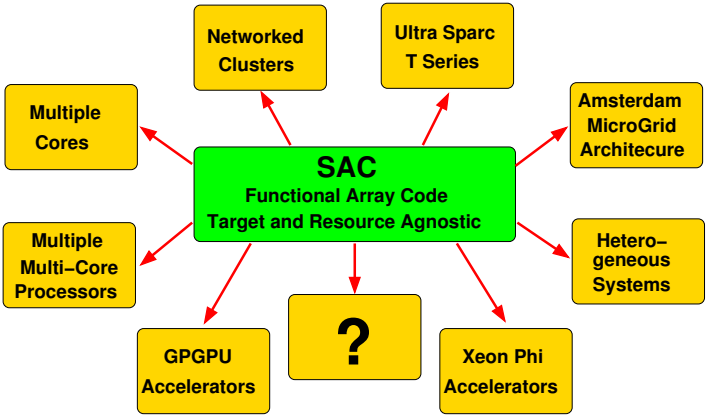
- memory management code
- parallelisation code
- synchronisation/communication code
- target hardware-specific arrangements
- ... and all the other clutter

Benefits:

- No race conditions
- No space leaks
- No deadlocks
- No .... you name it
- **Higher confidence in functional correctness**
- **Parallel execution for free**

Single Assignment C means ...

One source — many compilation targets:

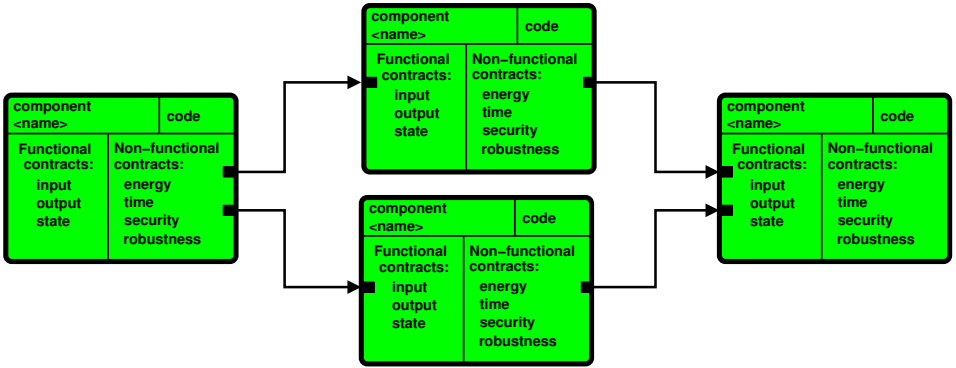


... and achieve reasonable performance for free !!

Coordination Language TeamPlay

Based on the concept of Exogeneous Coordination:

- Component identification
- Component interaction following **data-flow principle**
- Reasoning about **non-functional properties: time, energy, robustness, ...**
- Targeting cyber-physical systems



Heterogeneous Multi-core Coordination

Central question:

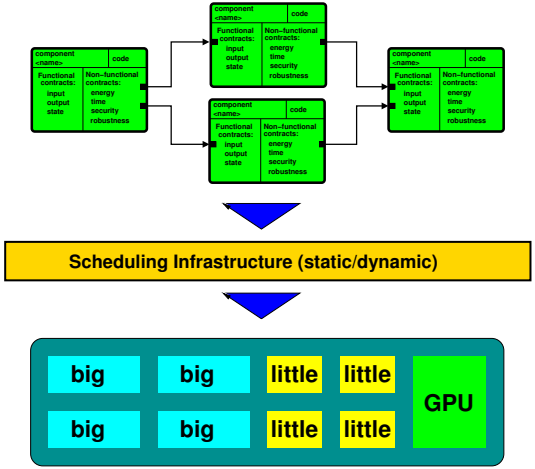
- **What to run where and when?**
  ... and why?

Constraints and objectives:

- Energy: want to minimize
- Time: must meet the deadline
- Maybe more ...

DVFS:

- Choose voltage/frquency pairs
- Per voltage/frquency island
- Change voltage/frquency over time



Energy Consumption Matters

Mobile devices:

- Battery-powered devices
- Less energy, more time to operate

Energy Consumption Matters

Mobile devices:

- Battery-powered devices
- Less energy, more time to operate

Sustainable computing:

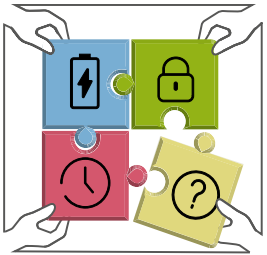
- Computing takes an ever increasing share of the world's energy resources.

Acknowledgments

ADVANCE
StatArch

apple core

TEAMPLAY



Time, Energy and security Analysis for
Multi/Many-core heterogeneous PLAtforms

ADMORPH

SUS Trainable