

High Productivity Programming Meets High Performance Execution

Clemens Grelck
University of Amsterdam
Systems and Network Engineering
c.grelck@uva.nl
staff.fnwi.uva.nl/c.u.grelck/



High Productivity Meets High Performance

The Problem: Multi-/Many-Cores rule the world:

- ▶ Different numbers of cores
- ▶ Different styles of cores
- ▶ Complex cache hierarchies
- ▶ General-purpose graphics cards
- ▶ Other accelerators: Xeon Phi
- ▶ Heterogeneous combinations of all the above
- ▶ Large variety of possible systems

High Productivity Meets High Performance

The Problem: Multi-/Many-Cores rule the world:

- ▶ Different numbers of cores
- ▶ Different styles of cores
- ▶ Complex cache hierarchies
- ▶ General-purpose graphics cards
- ▶ Other accelerators: Xeon Phi
- ▶ Heterogeneous combinations of all the above
- ▶ Large variety of possible systems

What does that mean for software ?

- ▶ All applications must become parallel
- ▶ Parallel programming tools geared to high performance computing experts
- ▶ Hard to use, error-prone, low productivity

High Productivity Meets High Performance

What does that mean for software ?

- ▶ All applications must become parallel
- ▶ Parallel programming tools geared to high performance computing experts
- ▶ Hard to use, error-prone, **low productivity**

Research questions:

- ▶ How can we make programming today's systems **easier for the average programmer ?**
- ▶ How can we make programming today's systems **more productive for the above average programmer ?**
- ▶ How can we still get **reasonable performance by aggressive compilation ?**

Single Assignment C and S-Net

SAC — Single Assignment C:

- ▶ Purely functional array programming language
- ▶ Very high-level programming style
- ▶ All resource management implicit
- ▶ Aggressively optimising large-scale compiler
- ▶ Automatic parallelisation for
- ▶ Check out: www.sac-home.org

Single Assignment C and S-Net

SAC — Single Assignment C:

- ▶ Purely functional array programming language
- ▶ Very high-level programming style
- ▶ All resource management implicit
- ▶ Aggressively optimising large-scale compiler
- ▶ Automatic parallelisation for
- ▶ Check out: www.sac-home.org

S-Net:

- ▶ Data flow coordination language
- ▶ Promotes almost normal functions to asynchronous components
- ▶ Arrangement of components into streaming network
- ▶ Adaptive runtime system maps computations to cores
- ▶ Check out: www.snet-home.org

Where to go from here ?

More information:

- ▶ Lecture in *Programming Concurrent Systems*
- ▶ Friday, December 12, 15–17, Science Park, B0.209
- ▶ Contact me: c.grelck@uva.nl
- ▶ Check out: staff.fnwi.uva.nl/c.u.grelck/

Where to go from here ?

More information:

- ▶ Lecture in *Programming Concurrent Systems*
- ▶ Friday, December 12, 15–17, Science Park, B0.209
- ▶ Contact me: c.grelck@uva.nl
- ▶ Check out: staff.fnwi.uva.nl/c.u.grelck/

Concrete projects:

- ▶ You name it !!
- ▶ I have ideas !!
- ▶ Depends on **your** prior knowledge !!
- ▶ Depends on **your** concrete interests !!

Where to go from here ?

More information:

- ▶ Lecture in *Programming Concurrent Systems*
- ▶ Friday, December 12, 15–17, Science Park, B0.209
- ▶ Contact me: c.grelck@uva.nl
- ▶ Check out: staff.fnwi.uva.nl/c.u.grelck/

Concrete projects:

- ▶ You name it !!
- ▶ I have ideas !!
- ▶ Depends on **your** prior knowledge !!
- ▶ Depends on **your** concrete interests !!

Prerequisites:

- ▶ Good programming skills, in particular C
- ▶ Parallel programming, depending on project
- ▶ Enthusiasm !!

Where to go from here ?

More information:

- ▶ Lecture in *Programming Concurrent Systems*
- ▶ Friday, December 12, 15–17, Science Park, B0.209
- ▶ Contact me: c.grelck@uva.nl
- ▶ Check out: staff.fnwi.uva.nl/c.u.grelck/

Concrete projects:

- ▶ You name it !!
- ▶ I have ideas !!
- ▶ Depends on **your** prior knowledge !!
- ▶ Depends on **your** concrete interests !!

Prerequisites:

- ▶ Good programming skills, in particular C
- ▶ Parallel programming, depending on project
- ▶ Enthusiasm !! **Publications ? YES, please !!**