# Simulation and Scientific Computing

## (Simulation und Wissenschaftliches Rechnen - SiWiR)

Winter Term 2015/16

Florian Schornbaum and Christian Kuschel
Chair for System Simulation

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

# Assignment 1: Performance Optimization

October 20, 2015 – November 9, 2015

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

TECHNISCHE FAKULTÄT

# Outline

- **Organizational Matters**
- Assignment 1: Performance Optimization
- Linear Algebra Algorithms
  - BLAS
  - ATLAS
- Performance Monitoring using likwid
- Remarks on using C++ for Assignment 1
  - Input & Output
  - Dynamic Memory Allocation (Generic Containers)

# Organizational Matters

- Don’t forget to sign up for the lecture and exercises on mein campus until Friday, October 30, 2015!
- Remember to sign up on the team formation list at our chair until Friday, October 30, 2015!
- Team formation is mandatory, you have to divide into groups of three persons.

# Organizational Matters

- Don’t forget to activate your **account** at our chair.
- Remote access to our computers (i10cip1.informatik.uni-erlangen.de to i10cip12) is only possible via **ssh** (Windows users: PuTTY & WinSCP).
- Access requires a **public-private key pair**:

```
[user@private-pc-at-home/laptop ~]$ ssh-keygen -t dsa -b 1024
```

Create a default key "~/.ssh/id_dsa" and copy (e.g., via mail) the public (!) key located at "~/.ssh/id_dsa.pub" to the "~/.ssh/" directory of your account at LSS. If the directory does not exist at your LSS account, you must create it. Then:

```
[user@i10cip5 ~]$ cat ~/.ssh/id_dsa.pub >> ~/.ssh/authorized_keys
```

- In order to login, you have to either first access a computer from the university network, or log in via VPN:

http://www.rrze.fau.de/dienste/internet-zugang/vpn/openvpn.shtml

# Outline

- Organizational Matters
- **Assignment 1: Performance Optimization**
- Linear Algebra Algorithms
  - BLAS
  - ATLAS
- Performance Monitoring using likwid
- Remarks on using C++ for Assignment 1
  - Input & Output
  - Dynamic Memory Allocation (Generic Containers)

# Assignment 1: Performance Optimization

⇒ let’s look at the assignment sheet

# Outline

- Organizational Matters
- Assignment 1: Performance Optimization
- **Linear Algebra Algorithms**
  - BLAS
  - ATLAS
- Performance Monitoring using likwid
- Remarks on using C++ for Assignment 1
  - Input & Output
  - Dynamic Memory Allocation (Generic Containers)

# BLAS

- **B**asic **L**inear **A**lgebra **S**ubprograms
- Routines that provide standard building blocks for performing basic vector and matrix operations [http://www.netlib.org/blas/]
- BLAS subroutines are a de facto standard API for linear algebra libraries and routines. (reference implementation: netlib)
- Many highly optimized implementations of BLAS exist for different computing languages and architectures.
- Often used in high performance computing
- BLAS functionality is divided into three levels:
  - Level 1: scalar, vector and vector-vector operations
  - Level 2: matrix-vector operations
  - Level 3: matrix-matrix operations

# ATLAS

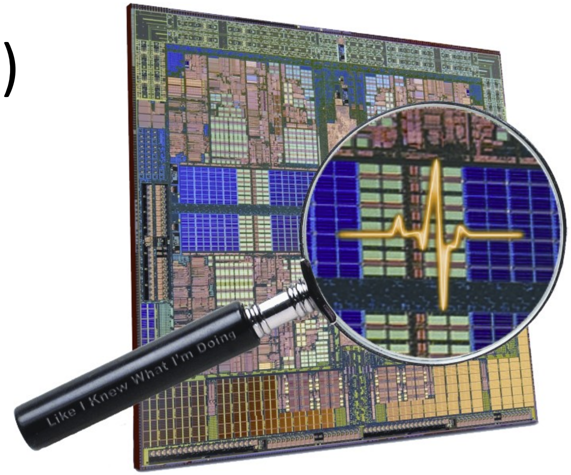
- Automatically Tuned Linear Algebra Software
- Highly optimized implementation of BLAS for C and Fortran77
- ATLAS makes use of Automated Empirical Optimization of Software (AEOS)
- During installation, ATLAS performs three optimization approaches of AEOS:
  - Parameterization: searching over the parameter space of a function (e.g., choosing the optimal blocking factor for best cache reuse, …)
  - Multiple implementation: searching through various implementations of the same function (standard C Code, SSE/AVX intrinsics, …)
  - Code generation: programs that write programs incorporating what knowledge they can about what will produce the best performance for the target system

# Outline

- Organizational Matters
- Assignment 1: Performance Optimization
- Linear Algebra Algorithms
  - BLAS
  - ATLAS
- **Performance Monitoring using likwid**
- Remarks on using C++ for Assignment 1
  - Input & Output
  - Dynamic Memory Allocation (Generic Containers)

# likwid

- **L**ike **I** **k**new **w**hat **I** am **d**oing (developed by Jan Treibig, RRZE Erlangen)
- Project providing tools to support programmers in hardware aware programming
- Lightweight and easy to use command line tools for Linux



- likwid-topology: information about architecture and caches
- likwid-perfctr: provides access to hardware performance counters that measure FLOPS, cache hit rates, and many more (marker API: allows to measure regions of the code)
- likwid-pin, …

# likwid

⇒ let’s look at an example (BLAS & likwid)

# Outline

- Organizational Matters
- Assignment 1: Performance Optimization
- Linear Algebra Algorithms
  - BLAS
  - ATLAS
- Performance Monitoring using likwid
- **Remarks on using C++ for Assignment 1**
  - Input & Output
  - Dynamic Memory Allocation (Generic Containers)

# C++ for Assignment 1

⇒ let’s look at examples/source code

# The End

## Questions ?

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

TECHNISCHE FAKULTÄT