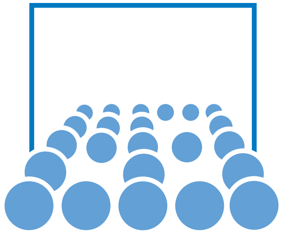
Seminar Resource-Aware Computing

Kick-Off Meeting

Chair of Scientific Computing (SCCS)

Summer 2015



Part I

Seminar Organization

Requirements

- **Language:** English
- **Independent literature research**
- **Paper:**
  - In total 5-10 pages (max 10 pages)
  - IEEE format
- **Peer review process:**
  - 2 reviews (topics will be assigned)
- **Presentation:**
  - 30 minutes talk
  - 15 minutes discussion
- **Mandatory attendance:**
  - Must attend all presentations
  - Rules for absence/sickness

Time Plan

Semester Week	Event
1st	Kick-off meeting
3rd	**Submission of paper outline. Flash Talk session.**
7th	Submission of paper draft to supervisors (voluntary)
8th	**Submission of paper for review**
9th	**Submission of reviews**
10th	Presentations (sessions 1 & 2)
11th	Presentations (sessions 3 & 4)
12th	Presentations (sessions 5 & 6)
12th	**Submission of paper final version**

Deadlines & Grading

Important Deadlines:

- **April 27th**, 23:59 PM: submission of paper topic and outline
- **June 5th**, 23:59 PM: submission of paper for review
- **June 12th**, 23:59 PM: submission of 2 reviews
- **July 3rd**, 23:59 PM: submission of final paper

Note: failure to meet these deadlines may result in grade deduction or failure.

Grading:

- Major components: final paper, presentation
- Deduction: attendance, deadlines, pre-release paper, plagiarism
- Bonus: commitment, activeness, self-implementation, etc.

Part II

Topic Overview

The Invasic Framework

Assigned to Martin Schwörer

- **Investigate the basics of invasion**
- **Specific to the Invasic project, or comparison with others**
- **For the whole thing to work, what components are needed?**
    - Applications
    - Communication btw. Apps and OS/RTS/RM
    - Language Support
    - OS/RTS/RM
    - Hardware cores/tiles
    - Communication amongst tiles
    - Monitor, security, power management, predictability, etc.

Resource-aware computing concepts

Assigned to Felix Scheffler

- **More general concepts, not specific to one project**
- **Different research groups/projects, different focus**
- **Different names/keywords:** resource aware, invasive, resource elastic, resource adaptive, etc.
- **Different approaches:**
  - Application initiates requests
  - System initiates requests
  - Application adapts to different resources
  - System takes care of data migration
  - . . .

Faults and Fault-Tolerance

Assigned to Christoph Hartlmüller

- **How to detect faults?**
- **How to react when fault occurs?**
- **Different forms of faults:**
  - Hardware failure: node level, core level, bit-level
  - Multiple faults, random faults
  - Time constraints: no results within time constraints
  - etc.
- **Solutions with different focus:**
  - By system design
  - Lauguage support
  - Hardware mechanism
  - Applications/algorithms

Invasive Algorithms

Assigned to Michael Riesch

- New computing paradigm requires algorithms that take the invasive computing techniques into account.
- Algorithms should be adaptive, able to adapt with changing resource requirements of the problem they are designed to solve.
- Capable of running concurrently and cooperatively on different hardware platforms.
- Capable of scaling with hardware constraints given from outside.

Application examples

- Image Processing
- Computer Vision
- HPC

Categorization of Problems and Extraction of Patterns

Assigned to Michal Szymczak

- Recurring techniques and solutions to problems may be condensed into algorithmic patterns
- Patterns are, in short, “a solution to a problem in a context”
- Using these, one can reason about classes of algorithms that apply the same patterns
- One may analyze that in regards to characteristic sizes, e.g. communication overhead and scaling behavior

Heterogeneous computing frameworks: OpenCL

Assigned to Oleksandr Shchur

- Initially developed by Apple, it is now an open standard maintained by the Khronos Goup
- Heterogeneous parallel computing standard
- Supports CPU, GPU and Accelerator devices.
- Features task parallelism as well as data parallelism

Resource Management for Heterogeneous Architectures

Supervisor: Isaias

Summary:

Heterogeneous systems are increasing in popularity in HPC. The current top systems in the top500 list of November 2014 contain nodes with Intel Xeon Phi and NVidia Tesla accelerators. It is anticipated that the use of accelerators will only increase. As a consequence, resource management software and algorithms need to be updated based on this trend.

Keywords:

- Accelerators
- Resource Management
- High Performance Computing

Invasive Network-on-a-Chip (iNoCs) Architectures

Supervisor: Isaias

Summary:

The number of cores integrated into integrated circuits is constantly increasing. CPUs in current HPC systems contain between 4 and 16 cores, and near future systems are expected to double or quadruple these core counts. With the increase number of cores, optimizations in hardware and software designs will focus on Network-on-a-Chip performance. The Invasive approach is of particular interest in this space.

Keywords:

- Network-on-a-Chip (NOC)
- Invasive Computing

Resource Aware Runtime Systems

Supervisor: Isaias

Summary:

Resource aware programming requires support from programming models, compilers, hardware and runtime systems. Runtime systems need to handle resource representation, performance modeling and scheduling efficiently. Current runtime systems and related algorithms need to be updated to efficiently handle application adaptations to changes in resource availability.

Keywords:

- Run-time (RT) system
- Resource Awareness

Dark Silicon

Supervisor: Josef

Summary:

The number of transistors available to chip designers has been increasing exponentially. In contrast, the power consumption per transistor has not. As a consequence, the number of transistors that will operate in future CPU designs will only be a subset of the total included in the design, because the power available per chip will remain constant. Hardware and software designs need to be adapted to this new power-bound reality.

Keywords:

- Power-bound
- Energy Efficiency

Power Saving by Dynamic Voltage and Frequency Scaling (DVFS) and Power Capping (PC)

Supervisor: Josef

Summary:

Dynamic Voltage and Frequency Scaling and Power Capping are features available in current CPUs. DVFS allows software to set the performance level at which the CPU operates, by setting the frequency and then a stable power lever on the chip. Power Capping allows the system to limit the maximum power in Watts that is used by the CPU. Software and tools need to be develop so that the best configuration is set so that energy efficiency and performance are optimized.

Keywords:

- Energy Efficiency
- Performance Level