Recent Advances in Parallel Programming
Seminar – SS 2017

Sascha Hunold and Jesper Larsson Träff

TU Wien

SS17
In this seminar, we will look at the state-of-the-art of languages, paradigms, and frameworks for efficiently programming parallel systems.
ECTS Types

- PL Programming Languages
- SE Software Engineering
- AL Algorithms
- TH Theory
“Seminars, where advanced new scientific material is discussed in a small group and students have to give presentations and (optional) have to write seminar papers; this kind of course should lead the students closer to actual scientific research.”

- from the Curriculum für das Masterstudium “Software Engineering & Internet Computing” an der Technischen Universität Wien (Gültig ab 1. Oktober 2015)

- optional means optional for the lecturer
What do you need to do?

- Select a topic (paper(s))
- Read and digest subject of paper (find related material)
  - incl. related work
- Present the topic in front of other students
- Write a seminar paper
- similar to BSc or MSc thesis
- but you “only” summarize work of others
- roughly 10-20 pages
  - there’s no minimum or maximum
  - it is done when it is done
  - depends on topics, number of graphics, etc.
Our Rules
Presence of each participant in all seminar presentations is obligatory.

Slides need to be sent to us at latest ONE week prior to presentation (one week is a lower bound)

We will add comments shortly after receiving the talk

Final slides must be sent to us (directly after talk)
Seminar paper due on 01.09.2017 (no extension)

optional: you may send us the paper (at latest 2 weeks before the deadline, such that we can add comments)
  - but we only comment ONCE
  - beware of summer holidays

You need to sign “Erklärung zur Verfassung der Arbeit” / statement of authenticity.
The number of meetings depends on the number of participants. There will be two talks per meeting.

- **Time:** 13:00 - 15:00
- **Paper selection:** March 30, 2017
  - and: How to give a seminar talk, write a seminar paper
- **First seminar talk:** April 27, 2017
Dates II

- dates (6 times)
  1. 13.4. (Easter vacations)
  2. 20.4. (Easter vacations)
  3. 27.4.
  4. 4.5. (EPILOG)
  5. 11.5.
  6. 18.5.
  7. 25.5 (holiday)
  8. 1.6. (conference)
  9. 8.6.
  10. 15.6. (holiday)
  11. 22.6.
  12. 29.6.

- max 12 talks!
You need to register on TISS until March 30, 2017.

- You should register with the seminar for which you want to have the ECTS points.
- That means you have committed. Dropping out afterwards will result in a failed course.
- after committing, all participants will be added to a TUWEL course
- there, we will upload a template for the seminar papers
- we also upload all the PDF files there
Topics

- IBM paper
- X10, PGAS (Partitioned Global Address Space)
- APGAS (Asynchronous PGAS)
- weak scaling analysis of different kernels
- IBM Power 775 supercomputer with 55,680 Power7 cores

- Charm++, cooperating message-driven objects (chares)
- demonstrate the practical applicability of the RTS features in mini-applications and real applications

- Charm++ and MPI
- “challenges in enabling interoperation between MPI and Charm++, and present techniques for managing the control flow and resource sharing ”
  - e.g., global comm, load balancing, libraries, etc.
- application studies

- problem: internode and intranode (shared-mem) communication
- extending MPI one-sided communication interface

- problem: MPI + X
- Regent, a high-productivity programming language
- “Regent programs look like ordinary sequential programs with calls to tasks."
- “compiler for Regent that translates Regent programs into efficient implementations for Legion, a dynamic, task-based asynchronous runtime system”
- evaluation with benchmarks

- UPC++, PGAS
- design of UPC++
- case studies with UPC++ (benchmarks)
- comparison to UPC and Titanium

- Coarray Fortran (CAF), PGAS
- investigate the capability of MPI in serving as the basis of a PGAS programming model such as CAF
  - was GASNet before
- evaluation with benchmarks
  - CAF + MPI vs. CAF + GASNet

- Kokkos, C++ library
- competitors: OpenMP, OpenACC, OpenCL
- benchmarks on Xeon, Xeon Phi, Kepler
- Kokkos executes computational kernels in fine-grain data parallel within an execution space.
- Computational kernels operate on multidimensional arrays residing in memory spaces.

- domain-specific languages (DSLs), code generation, Scala
- Delite framework for machine learning, data querying, graph analysis
- automatically parallelized and different parts of the application can run simultaneously on CPUs and GPUs
|---|---|

- code generation
- DSLs, pure-library embedded language
- Scala

- highly efficient cache-oblivious parallel recursive divide-and-conquer algorithms from inefficient iterative descriptions of DP recurrences
- identifies a recursive access pattern and a corresponding provably correct recursive algorithm for solving the DP recurrence
- autodiscovered algorithms significantly outperform parallel looping and tiled loop-based algorithms
Chapel
Chapel Background


A. Sidelnik, S. Maleki, B. L. Chamberlain, M. J. Garzarán, and D. A. Padua. “Performance Portability with the Chapel Language”. In: 26th IEEE International Parallel and Distributed Processing Symposium (IPDPS). 2012, pp. 582–594. DOI: 10.1109/IPDPS.2012.60
OpenSHMEM
M. Grossman, V. Kumar, Z. Budimlic, and V. Sarkar. “Integrating Asynchronous Task Parallelism with OpenSHMEM”. In: OpenSHMEM. 2016, pp. 3–17. DOI: 10.1007/978-3-319-50995-2_1

S. Jana, T. Curtis, D. Khaldi, and B. M. Chapman. “Increasing Computational Asynchrony in OpenSHMEM with Active Messages”. In: OpenSHMEM. 2016, pp. 35–51. DOI: 10.1007/978-3-319-50995-2_3
1. A. Bouteiller, G. Bosilca, and M. G. Venkata. “Surviving Errors with OpenSHMEM”. In: *OpenSHMEM*. 2016, pp. 66–81. DOI: 10.1007/978-3-319-50995-2_5


CnC concurrent collections

2. F. Schlimbach, J. C. Brodman, and K. Knobe. “Concurrent Collections on Distributed Memory Theory Put into Practice”. In: 21st Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP). 2013, pp. 225–232. DOI: 10.1109/PDP.2013.40