Universität Göttingen
Institute for Computer Science / GWDG
Julian Kunkel, Jonathan Decker

Project Overview

In order to obtain the credits, you will parallelize a non-trivial problem of your choice using the concepts and tools learned during the block course. As an alternative option, you could choose from an administrative topic that we derive from a practical problem at the GWDG.

First, you will decide upon a problem you like to solve, then you will create a sequential solution to this problem, and finally, you apply the experience of the block course to parallelize your application and analyze its scalability. You need to prepare a presentation for your fellow students as well as document your solution in a report. Both the presentation and the report are due at the end of the term and will be assessed and marked.

The successful participation in the project to be awarded with the university's credits consists of the following stages and rough schedule:

1. Block course: You completed the block course (01.04.2026 - 09.04.2026).
2. Team up: Form a group of 2-3 people[1] and discuss a rough idea/direction. See the list of example projects as well as projects from last year on our webpage for inspiration:
   `https://hps.vi4io.org/teaching/summer_term_2022/pchpc`
   `https://hps.vi4io.org/teaching/summer_term_2023/pchpc`
   `https://hps.vi4io.org/teaching/summer_term_2024/pchpc`
   `https://hps.vi4io.org/teaching/summer_term_2025/pchpc`
3. Assignment of supervisor: We will assign a supervisor per group.
4. Idea: Write up your project idea, approach and project plan. Discuss it with your supervisor. Create a short presentation. You should consider starting to work on the project a bit - create an early prototype to ensure you are able to pull it off. Discuss the idea with the supervisor as needed.
5. Project introduction meeting (see the webpage for the meeting date). Students present their project idea, approach and project plan. Talk about 5-7 minutes per student.
6. Coding: Create the sequential/parallel code to solve the issue. Overlap this phase with the evaluation and report preparation. Discuss with the supervisor as needed.
7. Evaluation: Run the code, evaluate the performance, analyze why the performance is as observed. Discuss with the supervisor as needed.
8. Result Meeting (see the webpage for the meeting date): Students present their approach and result in a second longer presentation talk about 20 per group. This presentation accounts for 30% of the grade.
9. Preparation of the report. Summarize your project, goals, approach, results, challenges, and the evaluation. Discuss with the supervisor as needed.
10. Submission of the report (until the last day of the term). The report accounts for the remaining 70% of the grade. You have to submit your source code along with your report. Optimally, the source code should be in a Gitlab or Github repository.

The report and the talks are shared between all team members. However, for the talk, each person must present a subset of slides and in the report, the roles of all members must be listed.

[1] It is possible to work alone on the project, however, there should be good reasons to do so.

Credits

Depending on whether you are a bachelor or master student and for what module you are going to make this course count, you might receive 5 or 6 credits. For master students, depending on the effort you are willing to put in this course, you can additionally choose between 6 and 9 credits. The course module M.Inf.1829 itself gives 6 credits and can be extended via module M.Inf.1834 for another 3 credits. Please note that M.Inf.1834 can also be taken to extend our course HPCSA and it can only be taken once. The more credits you request, the more strict we will be when evaluating your work. Make sure to state how many credits you request when handing in your final report, otherwise we will default to 6 credits.

Details about the content of the presentation and the report are described further down in the document.

Estimated by 30 hours per credit point, the amount of hours to be spent on the project are estimated as follows:

Credits	Block course	Supervisor interactions	Coding	Evaluation	Prep talk	Meetings[2]	Prep report
5	40	4	50	12	20	4	20
6	40	6	65	15	20	4	30
9	40	10	130	26	20	4	40

This is just a suggestion for your project plan to show how the available time can be distributed.

Task 1: Presentation - Project Introduction

For the project introduction, please include the following sections and information:

- Problem outline
  - Shortly describe the problem you are trying to solve, give context if possible.
- Project approach
  - Outline your approach to solving the problem.
  - How are you going to parallelize the program?
  - How can the problem complexity be increased to ensure we need to utilize parallel programming?
- Project plan
  - What are your major steps to solving the problem?
  - What do you need to develop?
- Project goal
  - What would success look like?

You are free to use any software you can agree on within your group to prepare your presentation. We suggest you use LaTeX to prepare your presentation. The GWDG provides a LaTeX environment *ShareLaTeX* that can be used by all students `https://sharelatex.gwdg.de`. Further, you can find presentation and report templates on our webpage `https://hps.vi4io.org/teaching/ressources/start#templates`.

[2]These are the scheduled meetings, not the time for meeting/discussing synchronously or (via email) with the supervisor.

Task 2: Coding

First, you should code a sequential version. Then, you should parallelize it using MPI. Lastly, you may parallelize selected regions using OpenMP. You can include additional libraries, but you should argue why they are required.

It is possible that your project might fail and you do not achieve the goal you set out to accomplish in your sequential program or you cannot resolve certain bugs. It is important to derive a small program that is actually executable - albeit it may not resolve the goals and evaluate it (see the next task). As long as you have something runnable and show qualitative work through your presentation and report and that you have understood the concepts and tools taught during the block course, you can still earn very good grades.

Hints

- Your SSH keys for the SCC will be valid for the entire duration of the course and can be used for your project as necessary.
- The command line editors nano and vim might be suboptimal for you to work efficiently.
- We recommend using a tool such as SSHFS to mount the code remotely. This way you can use any IDE or editor on your local machine while you edit data on the cluster directly. Some IDEs/editors are also able to copy files to a cluster.
- Please host the code you produce on GWDG Gitlab `https://gitlab.gwdg.de/` such that your supervisor can inspect your code. You may also include your presentation and report in a Git Repository or even the same as your code.

Task 3: Evaluation

It is mandatory to test and execute the code on the GWDG cluster and also evaluate its performance. Hence, a part of the presentation and report must be the analysis or evaluation of the performance your solution achieved. You should do a small theoretic analysis, then apply the concepts and tools you learned during the block course to practically evaluate the performance and behavior of your program.

The evaluation could include the following:

- Single process analysis of performance counters provided by a tool such as LIKWID or Vampir.
- Inclusion of a performance report / application specific workload by your application, e.g., doing X cycles/s.
- Graphs with strong-scaling / weak-scaling performance behavior for running on 1-10 nodes, vary process numbers/input.
- Several graphs from Vampir or a similar tool showing communication/performance behavior, possibly for multiple configurations of nodes.
- Discussion of the reasons behind the observations.

Task 4: Presentation - Project Results

This presentation should take about 20 minutes per group and will account for 30% of your grade. We will assess the presentation of each team member individually, but often this leads to very similar marks.

4.1 Presentation Outline

This is a template with an example outline for a presentation. Depending on the problem you worked on, not all points listed here might be applicable.

- Problem description
  - Description of the problem you are trying to solve
- Solution approach
  - What solution approaches did you consider and what did you go with?
- Sequential solution
  - Show your sequential solutions
  - Why is it problematic that it is only sequential?
- Parallelized solution
  - Explain your approach to parallelization
  - What elements of your application were eligible for parallelization?
  - Which of them did you chose to parallelize?
  - Why did you parallelize these parts of your application?
- Performance analysis / evaluation
  - Show the performance gain from applying parallelization
  - Runtime measurements
  - Graphs!
  - How does the application scale with problem size/number of processors (weak-scaling/strong-scaling)?
  - Further points as shown above
- Conclusion
  - What did you learn?
  - Did you accomplish your goal?
  - Contributions, overview of what you produced

Task 5: Report

The report should be at most 15 pages per group member without title, listings and appendix, from introduction to conclusion included. The report will account for 70% of your grade. You have to submit your source code along with your report. Optimally, the source code should be in a Gitlab or Github repository.

- The report must be submitted in PDF format.
- Access to the code repositories for the final submission must be given to Julian Kunkel and Jonathan Decker.
- Individual contributions of each group member must be described in the report appendix.

You should structure your report along the sample outline provided below.

5.1 Report Outline

This is a template with an outline for a final project report. Depending on the problem you worked on, this template might not be fully applicable. Adapt it as you see fit.

1. Introduction
    - Brief problem description and motivation
    - Aims and objectives of your project
    - Brief description of your approach
    - Quick summary of your most significant outcomes and their interpretations
    - Organization of the report
2. Methodology
    - Problem statement, Requirements specification
    - Solution approach
    - How is the simulation/parallel program modeled? What equations - if any - are used?
    - Sequential implementation design
    - Options for parallelization
    - How is the code distributed across nodes for distributed memory and for shared memory?
    - Expected improvements
    - Parallel implementation design
    - Experiment design, how did you validate the improvement
    - Include figures
3. Implementation
    - Discuss key implementation details of your solution.
    - Include relevant excerpts from your source code.
    - Is your code able to satisfy all functional requirements?
4. Performance analysis / evaluation
    - Show the concrete configurations you used in your performance analysis.
    - Show your observations from the program and performance evaluation.
    - Include a theoretic analysis and discussion of the results obtained.
    - How does the application scale with problem size/number of processors? Weak-scaling/strong-scaling?
    - Is the performance as you expected? Include some arguments and speculation for the reasons behind the evaluation results
    - What significant observations did you make?
    - How did you use the performance analysis tools?
    - Include figures and tables.

5. Challenges / Discussion
    - Mention the issues and challenges you faced and how you overcame them or if they are actually remaining (you cannot do everything).
    - What would you have done differently?
    - What other options to solving your problem should be mentioned?
6. Conclusion
    - Summarize the problem you set out to solve
    - Include your most important findings
    - Was your project successful?
    - What did you learn?
    - What did you achieve?
7. Appendix
    - Work sharing in the team - describe clearly who has done what.
    - May contain longer code snippets

We strongly suggest that you use LaTeX to work on your report but won't enforce it as long as you submit your report in PDF format. The GWDG provides a LaTeX environment *ShareLaTeX* that can be used by all students `https://sharelatex.gwdg.de`. Further, you can find presentation and report templates on our webpage `https://hps.vi4io.org/teaching/ressources/start#templates`.

Finally, submit your report, slides and links to code repositories via email to Jonathan (`jonathan.decker@uni-goettingen.de`). We would like to publish your results on our webpage, please mention in your submission email whether we may do so or what parts we may publish. You can revoke your consent any time via email and we will take your work down from our webpage. Refusing to let us publish your results has no influence on your grade.